CSCI-C 241 Lecture Notes

Erik Wennstrom

updated 2021/8/23

2021/8/23 2020 version imported

Introduction

These notes are meant to be a reference, covering the same material as in the lectures. They are not a good substitute for coming to class and taking your own notes.

1 Propositional Logic

Introduction: Cards and IDs

I'm going to start the semester with a demonstration adapted from a famous experiment in psychology[1]. Obviously, the results we get in the lecture are not going to be scientifically meaningful, but most semesters, we get results that are similar to those from the original study and its many replications.

Suppose you are presented with four cards on a table. One side of every card has a letter, and the other always has a number.

$$\boxed{8}\ \boxed{9}\ \boxed{S}\ \boxed{E}$$

You are shown a sentence and asked to think about whether it is true or false for the cards on the table:

> If there is an even number on one side of the card, then there is an "E" on the other side.

You are then asked the following question:

Question. Which cards (if any) do you need to turn over to determine whether the sentence is true or false?

[1] It's known as the "Wason Selection Task" or the "four-card problem", if you want to read more about it.

In the lecture, we did a quick little poll to determine what information people think they need in order to evaluate the truth of the sentence. If you're reading this at home, take some time to come up with an answer yourself before reading any further.

Okay, now that you've got your own guess, let's analyze this a little more closely:

The sentence is making a statement that some of these cards (the ones with even numbers on one side) meet a particular requirement (there's an "E" on the other side). The *requirement* is that there is an "E". But of course, we're not requiring that for *every* card. We only require it for cards that meet a particular *condition.* So we only need to check for the letter "E" on cards that have even numbers. In particular, the $\boxed{9}$ card doesn't have an even number on it, so we do *not* need to check the $\boxed{9}$ card.

But the condition of having an even number clearly applies to the card $\boxed{8}$, so we *do* have to flip that card over to make sure that it has the letter "E" on the other side. If it doesn't, then we know that the statement is false.

So far, so good. Now let's turn our attention to the cards that are showing letters. Let's start with the $\boxed{\text{E}}$ card. Do we need to flip it over? There are two possibilities.

Possibility 1: the other side has an even number on it. In that case, the rule does apply, and so the current side must have an "E" on it, which it does! So in possibility 1, the $\boxed{\text{E}}$ card conforms to the statement.

Possibility 2: the other side has an odd number on it. In that case, the rule doesn't apply, so it doesn't matter whether or not there's an "E" on this side of the card. So in this situation, the $\boxed{\text{E}}$ card still conforms to the statement.

Since the number on the other side can't change whether the statement is true or false, we don't need to flip over that card.

Finally, let's take a look at the $\boxed{\text{S}}$ card. Again, let's look at the two possibilities.

Possibility 1: the other side has an even number on it. In that case, the rule does apply, and so the current side must have an "E" on it, which it doesn't. So if we're in this case, we know that the statement is false.

Possibility 2: the other side has an odd number on it. In that case, the rule doesn't apply, so it doesn't matter whether or not there's an "E" on this side of the card. In this situation, the $\boxed{\text{S}}$ card won't help you disprove the statement.

Since there's a difference between the truth of the statement, we really do need to flip over the $\boxed{\text{S}}$ card to see whether or not it disproves the statement.

If you didn't get it right, don't feel bad. This test has been used in many psychological experiments, and the first time it was run, fewer than 10% of the people tested answered correctly. But when the logic is explained afterwards, most people accept the answer I gave as correct.

There are a lot of different theories as to why humans perform so badly on problems like this, so I'm not going to make any broad claims about psychology here. But I will mention one interesting twist. If we change the context from abstract rules about cards with letters and numbers (or cards with numbers and colors, or anything abstract and novel), people often do much better. In particular, when we change the context to social rules, people seem to be very good at determining the truth of conditional ("if-then") statements like this one.

For example, suppose you're working at a bar and it's your job to make sure the following law is upheld:

> If a customer is drinking alcohol, they must be at least 21 years old.

You just came back from your break and there are four people in the bar. There are two new people at the near end of the bar. You remember their names, but not how old their are. Lee is drinking a vodka tonic and Kim is drinking a Dr Pepper. At the other end of the bar are two regulars whose IDs you checked earlier. You remember that Maria is 62 years old and Jess is only 17. You can't tell what they are drinking from where you're standing.

So do you need to check the IDs of Lee and/or Kim? Do you need to go peek at what Maria and Jess are drinking? Make sure you have answers for all these questions before moving on.

Almost everybody gets these right. Obviously, you check Lee's ID because they meet the alcohol-drinking condition, and thus they also need to meet the over-21 requirement. But it's almost as obvious that you don't need to see Kim's ID because they're only drinking a Dr Pepper. Kim doesn't meet the condition, so they automatically satisfy the rule. What about Maria? Well you know she meets the age requirement, so it doesn't matter if she's drinking alcohol or not. But Jess might or might not be breaking the rule, so we need to look at what they're drinking.

If you think about it, this is the exact same situation as earlier. We just changed some basic ideas around. Cards were replaced with people, having an even number was replaced with drinking alcohol, and having the letter "E" was replaced with being at least 21 years old.

It's still not clear exactly why we're better at reasoning about social rules then we are about cards, but it is clear that our intuitions about even very simple logical statements are not very reliable when we're presented with new situations.

In our day-to-day lives, we're used to being able to just read a statement and understand exactly how to tell if it's true or not without even thinking about it. We do this intuitively all the time, without any kind of conscious analysis. But since this intuition is not always reliable, we should sometimes turn a critical eye upon reasoning itself. And this is why I think it's important to study logic itself in a formal, systematic way.

Fortunately, there are situations in which our intuitions *are* reliable. If we can identify those situations and use them as metaphors to help us learn the rules of logic. Once we learn what those rules actually are, instead of just relying upon our intuitions, we can apply those rules to any situation at all, no matter how weird.

Once you get some practice working with conditionals, you won't need to think about underage drinking just to make sure your logic is sound. Eventually, you'll just understand the rules of conditionals without having to think about them. But while you're still learning, metaphors can be very useful tools, *as long as you pick the right context for each problem.*

1.1 Propositions

The word **proposition** is usually given a definition somewhere along the lines of "a statement that can be either true or false". I don't think that this is a *bad* definition, but I think it can cause confusion in some students. First of all, it's redundant, but it's not obviously redundant. The definition could be shortened to just: "a **proposition** is a statement," or to "a **proposition** can be either true or false." After reading this definition, some students end up thinking that a proposition is a special kind of statement.

Sometimes students think that only statements that have changeable truth values count as propositions (ruling out logical truths, mathematical tautologies, and contradictions). But statements like "$1+1=2$" and "The sky is both blue and not blue," are propositions.

Occasionally people get the idea that propositions must have truth values are objectively knowable count as propositions (ruling out things like opinions or statements about the future). Again, this is a misconception. Claims like "Bigger is better," and "It will rain tomorrow," are in fact propositions.

The thing to keep in mind is that the word "proposition" is a *grammatical* definition, or as computer scientists (and logicians) would say: it's a definition based on *syntax*. It's about the *format* of the expression, not the *meaning* of the words or symbols involved. Any statement, claim, or declarative sentence is a proposition because it *makes sense to talk about* whether they are true or not.

So here's my definition of a proposition.

Definition 1.1. If it makes sense to talk about whether an expression is true, then it is a **proposition**. If it makes sense to talk about whether it is false, then it is a **proposition**.

So vague declarative sentences ("This is an expensive item."), claims about unsolved problems ("The Goldbach Conjecture is false."), and even (arguably) paradoxical statements ("This sentence is false.") all count as propositions. Some of them are really dumb propositions, and we're usually interested in the

This raises an obvious question: What *doesn't* count as a proposition? Even if we rule out silly answers like the number 5 and stick to English sentences, there are quite a few possibilities.

For example, questions (e.g., "What is the air-speed velocity of an unladen swallow?" or "Do you know the way to San Jose?") are not propositions. Of course yes/no questions are very closely related to propositions, but they aren't *technically* propositions. A lot of things that are true for propositions are also true for yes/no questions, but not everything. For example, it doesn't make sense to talk about whether "Are you ready to rumble?" is true or false, but you can talk about whether the related proposition “You are ready to rumble,” is true or false.

Similarly, commands, instructions, and other imperative statements aren’t technically propositions, although they are closely related to propositions. It wouldn’t make sense to say that “Take me out to the ball game,” is true (or false), so that isn’t a proposition, but it’s closely related to the proposition “You will take me out to the ball game,” or to the proposition “You should take me out to the ball game.”

Of course sentence fragments like “All the king’s horses,” aren’t propositions either. But there are some noun phrases that are very closely related to propositions. For example, “The presence of a chaperone at the dance,” is closely related to the proposition “A chaperone is present at the dance.” Similarly, “The dependence of a fixed-point instruction on a value loaded from memory,” (noun-phrase; not a proposition) is closely related to “This fixed-point instruction depends on a value loaded from memory” (declarative sentence; proposition).

And there are a smattering of other types of non-declarative utterances that wouldn’t count as propositions, including interjections (“Drat!”) and certain kinds of social niceties (“Good morning.”)

1.1.1 In Mathematical Notation

Propositions are not limited to natural languages. There are lots of perfectly good propositions that are best written using mathematical notation. So for example, “$2 + 2 = 3$”, “$5 \geq 4^2 - 8$”, and “$2 \mid n^2 + n$ for all integers n” are all propositions. (The first one is false, and the other two are true.) When it comes to mathematical statements, we don’t have to worry about fuzziness or opinions, but we might have to worry about context in order to determine whether it is true or false. We still think of an equation like $x^2 + 3x - 5 = 0$ as a proposition, even though its truth depends upon which value of x we use. Any equation or an inequality is either true or false (as long there’s enough context), so any equation or inequality will always a proposition.

Not every mathematical expression is a proposition, however. For example, consider the polynomial $x^2 + 3x - 5$ (notice the lack of an equal sign). Even if you specify what x is, you won’t end up with a claim that is either true or false. Instead, you’ll end up with some number. Similarly, $3^4 - 11$ refers to some number, not some claim that can be agreed with or denied. Just like in English, this is a matter of grammar (syntax). Both of these expressions are essentially noun phrases without any verbs. An expression needs some kind of verb in order to be a proposition.

1.1.2 In Programming Languages

To some extent, you can even talk about propositions in programming languages, although we usually use different terminology there. In a purely functional programming language a command that returns a boolean value is basically the same thing as a proposition. Of course, once you start getting into side effects, the analogy gets a lot weaker.

1.2 (Symbolic) Propositional Logic

If we take simple propositions and connect them together using logical words like “and”, “or”, “not”, or “if”, we can build up more complex propositions, which we call **compound** propositions. Those little words like “or” and “if” that we use to combine propositions are called **connectives**. A proposition that can't be broken down and reduced to combinations of other propositions is called an **atomic proposition**.

In many cases, once you know whether the smaller propositions are true or false, the truth of the compound proposition can be determined by simple, rigid rules.

These are *logical* constructions because the truth of the compound composition can be determined by looking at the truth of the simpler propositions. For example, if we know whether the propositions “‘the subject has used an iPod before” and “they read the manual” are true or not, we can use those facts to determine the truth of the sentence “Either the subject has used a iPod before, or they read the manual.”

The study of propositional logic is the study of how the truth of simple propositions affects the truth of compound propositions. When we study propositional logic, we don't actually care about whether the atomic propositions are true or not. As logicians, all we care about is how the truth value of the atomic propositions is connected to the truth value of larger propositions.

To help us ignore what needs to be ignored, it's helpful to get away from messy, ambiguous natural languages (like English), and to create a precise language to help us talk about these ideas. The precise language that logicians created to talk about propositional logic is called **symbolic propositional logic**. We often use the phrase “propositional logic” to refer to this specific symbolic language instead of using it to refer to the entire field of study. You can usually figure out which meaning is apporpriate from context.

Let's talk about the basics of the language of propositional logic. Atomic propositions are represented by atomic propositional formulas (which are just letters, like P, Q, or R).[2] Compound formulas are built up by connecting other formulas with connectives (like $\wedge$ and $\vee$). Each connective connects the truth values of the atomic formulas to the truth of the entire formula in a particular way.

1.3 Conjunction

Perhaps the most simple way to combine two propositions is to use the word “and”. For example, the proposition “The door is locked, and I forgot my

[2] We usually use capital letters from the English alphabet to stand for atomic propositions. Most of the time, when you see lowercase letters, you're looking at a variable that can stand in for any proposition (whether it's atomic or compound).

keys," is true when both atomic propositions ("The door is locked," and "I forgot my keys,") are true, and it is false in every other circumstance. This kind of connection is called **conjunction**. Strictly defined, a conjunction is true when both parts (we call them **conjuncts**) are true and it is false when either or both of the conjuncts is false.

In the version of propositional logic that we're using in this class, we use the symbol $\wedge$ to stand in for conjunction. This symbol is sometimes called "wedge" or "meet", but when we read it out loud as part of a formula, we usually just say "and." Some systems might use an ampersand & or a multiplication dot $\cdot$. In programming languages, you'll often see conjunction represented as a boolean operator by an ampersand `&`, a double ampersand `&&`, or just the word `AND`.

If you are hand-writing the symbol, think of it as a pointy version of the intersection symbol $\cap$. If you are typing the symbol, note that it is *not* the same as the caret symbol ˆ. In LaTeX, the symbol is `\wedge` (Unicode: `U+2227`, HTML: `&and;`).

Now we can get away with defining the concept of conjunction ($\wedge$) using the English word "and" because "and" is a pretty stable word in English and it almost always has the exact same meaning. But that's not going to be the case with all of our connectives, so instead of relying upon English to define our connectives, we can instead write a definition by using a table:

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

A table like this is called a **truth table**. Each row corresponds to a particular assignment of "true" or "false" to each of the smaller propositions. We call this combination of choices of truth or falsehood a **truth assignment** (or sometimes just an **assignment**).

The first row corresponds to the truth assignment where both p and q are true. In that case, since both are true, we defined the conjunction $p \wedge q$ to also be true. We say that the truth assignment $(p = \mathrm{T}, q = \mathrm{T})$ **satisfies** the formula $p \wedge q$. New students will often say that the assignment "makes the formula true", which isn't wrong, but "satisfies" is more formal and less likely to be misunderstood.

The second row corresponds to the truth assignment $(p = \mathrm{T}, q = \mathrm{F})$ (where p is true and q is false), and under that truth assignment, the formula $p \wedge q$ is false. There's no special verb for describing when a formula is false under some assignment. We just say that the truth assignment "doesn't satisfy" the formula.

When you are expressing a conjunctive proposition in English, the most natural word to use is "and", perhaps accompanied by a word like "both" if needed for clarity. Occasionally, you might use a different word, such as "plus", as in "I don't like the taste, plus it gives me gas."

There might also be an extra word or phrase like "as well", "also", or "too", which are often added to emphasize the conjunction, but they are not required.

In most natural languages, it's possible to express a conjunction without using any logical words at all. If you just write two declarative sentences, one after the other, then the usual meaning is that you are claiming both sentences are true. In other words, it's a conjunction. Similarly, if someone combines two sentences with a semicolon, they are typically making a conjunctive claim. The sentences "She has black hair, and he does not," and "She has black hair; he does not," have the same meaning, as does the short paragraph "She has black hair. He does not."

1.3.1 Disjunction

Consider the word "or," as it is used in sentences like "The subject has either used the device before, or they have read the manual." This is what's we calle a **disjunction**. To get a feel for how disjunction works, let's examine each of the four possible truth assignments for this example.

If we have a subject who has used the device before but not read the manual (the first part is true and the second is false), then this sentence would be true. Also, if we had a subject who had not used the device before, but who *had* read the manual (the first part is false and the second is true), then the sentence would also be true. If the subject had neither used the device before nor read the manual (both parts are false), then the sentence is clearly false.

The last possibility is when we have a subject who has both used the device before and who has *also* read the manual. Whether this should make the original sentence true or false depends a little bit upon the context.

If this statement was a conclusion that an observer made based on the fact that the subject seemed to know what they were doing, then the observer probably meant to include the possibility that both parts were true. This meaning of the word "or" (either one or the other or both) is called **inclusive** disjunction.

Here's a truth table definition of inclusive disjunction:

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

To put it another way, an inclusive disjunction is true when either or both parts (called **disjuncts**) are true. But that's not the only way to interpret the word "or".

For example, if the speaker was running an experiment to test whether reading the manual was more or less effective than practical experience, then this sentence might be intending to rule out the possibility that the user has both read the manual and used the device before. This interpretation of the word "or" (where exactly one of the disjuncts is true, but not both) is called **exclusive** disjunction.

Here's a truth table definition of exclusive disjunction:

p	a	$p \oplus q$
T	T	F
T	F	T
F	T	T
F	F	F

In technical settings, inclusive disjunction is more common, and so when you see the word "disjunction" by itself (without the word "exclusive"), you should assume that it means inclusive disjunction. In fact, most of the time in this class, when you see the word "or", you should assume the inclusive meaning, unless there are strong contextual clues implying otherwise. For example, in a sentence like "Your meal comes with a side order of fries or onion rings," you can probably assume that you can't get *both* fries and onion rings, so it would definitely be reasonable to treat this as exclusive disjunction.

In this class, we will use the symbol $\vee$ to represent (inclusive) disjunction. The symbol is sometimes called "vee", "vel", or "join", but we usually just say "or" when reading it out loud. Occasionally, you might come across a logical system where disjunction is represented by the plus sign $+$. In programming, the boolean operator for disjunction is often written using a vertical bar `|`, two vertical bars `||`, or just the word `OR`.

If you're writing the symbol by hand, think of it as a pointy version of the union symbol $\cup$ from set theory. If you are typing the symbol, do not use the letter `V`. This is a different symbol, which you can write using `\vee` in LaTeX (Unicode: `U+2228`, HTML: `&or;`).

In ordinary English, disjunction is most commonly represented by the word "or" (often accompanied by "either"), but since the inclusivity or exclusivity is heavily dependent on context, people often add "or both" to make it clear that inclusive disjunction is intended. Another way to make it extra clear that the inclusive disjunction is intended is to use "and/or". (Is that surprising to you? If so, try to build the truth table for a sentence like "You can have pepperoni and/or mushrooms," and see where that gets you.)

We won't talk much about exclusive disjunction, but if you're curious, the most common symbol used is $\oplus$ (LaTeX: `\oplus`, Unicode: `U+2295`, HTML: `&CirclePlus;`). Occasionally, you might see the symbol $\veebar$. In programming, exclusive disjunction is often just left out of the boolean operators, but if it is included, it is usually written `XOR`, `EOR`, or `EXOR`.

If you want to express an exclusive disjunction in English without any ambiguity, you can say something like "…or …, but not both."

1.4 Negation

The next connective we're going talk to about is called **negation**, and it has roughly the same meaning that the word "not" does in English. We call it a "connective", but instead of connecting two formulas together, negation simply modifies a single formula. The symbol we'll be using for negation is $\neg$. Think of it like a minus sign with a little hook on the end. In LaTeX, the code is `\neg` (Unicode: `U+00AC`, HTML: `&not;`).

For any formula p, the formula $\neg p$ is true exactly when p is false, and it is false exactly when p is true. In tabular form:

p	$\neg p$
T	F
F	T

Translation can be deceptively tricky for negation. When it comes to conjunction, simply sticking the word "and" between two English propositions is pretty much the same thing as sticking the symbol $\wedge$ between their corresponding propositional formulas, and disjunction works pretty similarly. But you can't just stick the word "not" at the beginning of any sentence and expect it to make sense. For example, the negation of "He is tall" can't be written "Not he is tall". You could write its negation as "It is not the case that he is tall", and that's fine as long as you don't mind sounding like a robot. The most natural way to negate this sentence would be: "He is not tall."

NOTE: It would not be appropriate to use "he is short" as the negation of "he is tall." (What if he was of average height?) Don't confuse negation with opposites.

When it comes to negating sentences, it doesn't always boil down to finding a sensible place to stick the word "not". There are lots of places you can insert a "not" that wil result in a grammatically correct sentence, but not the *correct* sentence. For example, if you someone told you "There exists an even number that is also odd," and you disagreed with them, you might say "It is not true that there exists an even number that is also odd," or you might say "There does not exist an even number that is also odd," or even "There are no even numbers that are also odd." But it would be wrong to say "There exists an even number that is not odd." (Well, that's a true sentence, but it doesn't negate the original sentence.)

Example 1.1. For practice, try to represent the negations of the following English propositions in a natural way.

- "My dog ate my homework."
- "George is always late."
- "I am not a crook."
- "No man is an island."

Here's are some possible negations.

Solution.

- "My dog didn't eat my homework."
- "George is not always late." (or maybe "At least some of the time, George is not late.")
- "It's not true that I am not a crook." (or less confusingly, "I am a crook.")
- "It's not true that no man is an island." (or maybe "At least one man is an island." It would be wrong to write "Every man is an island.")

1.5 “Translating” between English and Logic

Example 1.2. Translate the sentence “I finished the project, and I didn’t make a mistake.” into propositional logic.

If you want to get technical about it, it’s not truly possible to translate an English sentence like this into a formula of propositional logic. There’s no way to talk about projects or mistakes or finishing things in propositional logic. Heck, there isn’t even a concept of “I”. So when we ask you to “translate” from English to symbolic logic, what we’re really expecting you to do is to make up some short definitions for the atomic propositions, and then translate the logical parts of the sentence using the appropriate symbolic connectives.

So if you’re doing a translation problem and you aren’t given any definitions for the atomic propositions, your first step needs to make up your own definitions for the propositional variables. Here’s one good translation of our example sentence:

P: “I finished the project.”
M: “I made a mistake.”

Translation: $P \wedge \neg M$

Make sure the propositions are as small as possible. Consider the following alternative translation for our example.

P: “I finished the project.”
D: “I didn’t make a mistake.”

Translation: $P \wedge D$

While this second translation isn’t *wrong*, the first one is definitely better as it captures more of the sentence’s meaning with logical symbols.

Here’s another example translation.

Example 1.3. “I washed the floor, and also the windows.”
F: “I washed the floor.”
W: “I washed the windows.”
Translation: $F \wedge W$

Notice how both definitions can stand on their own as English propositions. It would not be appropriate to define W: “the windows.” This may sometimes require filling in words, slightly altering the grammar, or replacing pronouns with what they are referring to.

1.6 “But”

How would you translate the sentence “The air is cold, but the water is warm,” into propositional logic? Do we need a special connective for “but”?

Think about how to tell whether the sentence is true or false, based on the truth or falsehood of its parts. If both parts are true (the air is cold and the water is warm), then the whole sentence is in fact, true. If the air is not cold, then the whole sentence is false (regardless of whether the water is warm or not). If the water isn't warm, then again, the whole sentence is false. We can summarize this in a table:

p	q	p, but q.
T	T	T
T	F	F
F	T	F
F	F	F

Look familiar? This is exactly the same table as conjunction $\wedge$. That means we should translate "The air is cold, but the water is warm," the same way as we would translate the sentence "The air is cold, and the water is warm."

If that still feels wrong to you, it may be useful to pin down exactly what it is that seems to be missing from the "and" version of the sentence. What does "but" communicate that "and" does not?

The difference is that "but" expresses some kind of *contrast* between the two parts. When we use the word "but," we are claiming that both parts are true, but we are also expressing that there is something different about the two parts of the sentence. Maybe one part is surprising given the other part ("It tasted bad, but I ate it anyway,"), or maybe the two parts are just expressing things that are somehow contrastive ("This little piggy went to market, but that little piggy stayed home.")

So there is a difference between "and" and "but", but it's not a *logical* difference. Using "but" instead of "and" doesn't affect whether the sentence is true or false. It's just a matter of connotation and emphasis. Since it doesn't affect whether the sentence is true or false, it makes sense that we would use the same symbol $\wedge$ to translate both "and" and "but" into propositional logic.

Of course, there are lots of words in English that are very similar to "but" that should also be translated using $\wedge$: "yet," "however,", "nevertheless," etc.

1.7 Conditional Implication (if)

We're now ready to turn our attention back to the example we started with: conditional implication. The word "if" often gets used in situations that have lots of subtle connotations, implied presuppositions, and other extra contextual details. I could easily fill an entire book just talking about the subtle differences between material conditionals, counterfactual conditionals, predictive conditionals, relevance conditionals, causal conditionals, strict conditionals, anankastic conditionals, and the like. But fortunately, there's an awful lot we can say about conditionals without getting into all of those messy details.

Let's consider an example of a conditional statement:

> "If the record stores a zip code, the record's length is 5."

A **conditional** sentence has two parts. The **premise** (also called the **con-**

dition or **antecedent**) is often[3] part of a dependent clause which starts with a conjunctive word or phrase like “if” or “in the case that”. In our example, the premise is “the record stores a zip code”. The premise describes the circumstances in which the main part of the sentence is being asserted. The main part (“the record’s length is 5” in our example) is called the **conclusion** or **consequent**, and is typically an independent clause. It’s hard to break down the meaning of the word “if” without making a circular argument, but if you twisted my arm, here’s how I would sum it up:

> When someone makes a conditional statement, they are telling you that under certain circumstances, a particular claim must be true. The **conclusion** tells you what that claim is, and the **premise** tells you what those circumstances are.

As students of propositional logic, we want to capture the essence of this concept by deciding which truth assignments should satisfy the conditional statement. In other words, if I know whether the premise is true or false and whether the conclusion is true or false, what does that tell me about whether the whole conditional statement is true or false? To put it a third way, what does the truth table for implication look like?

Let’s use the above example to try and analyze what such a truth table should look like. We’ll break down all four possibilities and see what we get.

The easiest case to analyze is the situation where the premise is true (the record stores a zip code) and the conclusion is false (the record does not have length 5). If you have a record like this, then it’s obvious that the sentence “If the record stores a zip code, the record’s length is 5,” is false. This would be a clear counterexample.

Now if the premise is true and the conclusion is also true (we have a record storing a zip code that *does* have length 5), can we conclude that the original statement is true? Yes! Well, sort of true. . . I mean we certainly can’t conclude that the sentence is *false*. But really, this is just one example record. What if there’s some other record that does *not* fit the requirement?

Unfortunately, propositional logic is not equipped to fully handle the distinction between a claim that is true for *all* records and a statement that is true about *one particular* record. And this is where we’re forced to make a compromise. Words like “if” automatically imply that we are talking about *all* of something and not just one particular thing.[4] And until we get to first-order logic, we won’t be able to fully capture this part of conditional statements.

So as long as we’re stuck with the limited tools of propositional logic, we will have to pick “true” or “false” based solely on one example. When we have a situation that does not serve as a counterexample, we’re going to pick “true”. So because a zip-code-storing record that has length 5 does satisfy the sentence

[3] Obviously, what I’m saying here is about conditionals in English, but things are pretty similar in other languages too.

[4] Even if it seems like we’re talking about just one particular situation (e.g., “If it rains tomorrow, my car will get wet.”), it turns out that we’re really talking about all *possibilities* for that situation.

"If the record stores a zip code, the record's length is 5," we are going to call that conditional sentence "true" for that particular record.

Let's turn back to our example, and try to keep in mind this broader idea of what it means for a conditional to be "true" in a particular situation (i.e., the situation isn't a counterexample). In both of the remaining cases, the premise is *false* (meaning we *don't* have a record that stores a zip code. And in these situations, it simply doesn't matter whether the conclusion is true or not. When both the premise and conclusion are false, we call the conditional "true". The presence of a telephone-number record with length 10 does not contradict our conditional sentence. When the premise is false and the conclusion is true, we still call the conditional "true". The presence of a non-zip-code record that has length 5 (maybe a name?) does not contradict the sentence 'If the record stores a zip code, the record's length is 5."

With this in mind, we can now build a truth table for conditional implication.[5]

p	q	$p \to q$.
T	T	T
T	F	F
F	T	T
F	F	T

The symbol we're using in this class for implication is just an arrow pointing to the right → (LaTeX: `\rightarrow`, Unicode: `U+2192`, HTML: `&rarr;`). We always put the premise on the left and the conclusion on the right. Note that you can*not* use a leftward-pointing arrow, even if you move the premise and conclusion: $q \leftarrow q$ isn't the same thing as $p \to q$.[6] In other logical systems, you might see a double-arrow ⇒ or the so-called "horseshoe symbol" ⊃.

If you take a look at that table, you'll notice that there's only one truth assignment that doesn't satisfy the conditional, and that's the one that satisfies the premise, but not the conclusion. In fact, this is one of the most important principles of this entire course: IN ORDER FOR A CONDITIONAL STATEMENT TO BE FALSE, THE PREMISE MUST BE TRUE AND THE CONCLUSION MUST BE FALSE. This fact will crop up all over the place, so make sure you've got it down pat.

1.7.1 Conditional Sentences in English

The classic translation of $p \to q$ into English is "If p, then q." Note that the word "then" is entirely optional here. Also note that the order in which the phrases appear doesn't matter. The sentences "If you open the door, the cat will try to get out," and "The cat will try to get out if you open the door," have the same meaning. You can tell which part of the sentence is the premise because that's the part that starts with "if". I like to think of "if" as some sort of label or tag

[5] If you want to get really picky about things, the type of simplified conditional we're talking about here is called the **material** conditional.

[6] This is because there are some logics that use the left-pointing arrow to mean a subtly different thing.

that is used to identify the premise of the conditional.

When translating an English conditional into a formula of propositional logic, try not to think about the *order* of the sentence. Instead, look for the word(s) that indicate that a particular phrase is the premise. In addition to "if", there are lots of other words and phrases that can be used to identify the premise of a conditional. Here is a (very incomplete) list of "tags" that identify premises in a way similar to the word "if":

- if ____
- given that ____
- in the case that ____
- provided that ____
- so long as ____
- when ____
- where ____
- whenever ____
- in the situation where ____
- should ____
- anytime ____ occurs

Example 1.4. Translate the following sentences into propositional logic.

(a) "Where there's smoke, there's fire."

S: "There's smoke."
F: "There's fire."

In this sentence, "where" identifies the clause "there's smoke" as the premise of a conditional.

Translation: $S \to F$

(b) There will be a small fee, should breakage occur.

F: "There is a small fee."
B: "Breakage occurs."

In this sentence, the word "should" in the phrase "should breakage occur" tells you that this clause is the premise.[7]

Translation: $B \to F$

[7] When "should" is used as an auxilliary verb (as in "You should do your homework."), it doesn't indicate that the sentence is a conditional. It only creates a conditional when it starts a clause as in the above example.

(c) You can return the product, provided that the seal has not been broken.

R: "You can return the product."
B: "The seal is broken."

Notice that I haven't included "not" in the definition of B. This is because I'm going to translate that using the symbol $\neg$.

Translation: $\neg B \rightarrow R$

Premise tags aren't the only way to express implication. One other method is to use a word that describes how the conclusion *follows* from the premise, or how the premise *leads to* or *implies* the conclusion. Here is a short list of some of the words and phrases that work like this.

- $\underline{\text{hypothesis}}$ implies $\underline{\text{conclusion}}$
- $\underline{\text{hypothesis}}$ leads to $\underline{\text{conclusion}}$
- $\underline{\text{conclusion}}$ is implied by $\underline{\text{hypothesis}}$
- $\underline{\text{conclusion}}$ follows from $\underline{\text{hypothesis}}$
- $\underline{\text{conclusion}}$ is deducible from $\underline{\text{hypothesis}}$

Grammatically, these structures are a little different from the premise tags we talked about earlier. For example, consider the sentence "Where there's smoke, there's fire," that we were talking about earlier. If we wanted to use "implies" to paraphrase this sentence, we would *not* say "There's smoke implies there's fire." That would be bad grammar. Instead, we need to take the individual sentences "There's smoke," and "There's fire," and convert them into noun phrases. So we *could* say "The presence of smoke implies the existence of fire," or even "The fact that there is fire is implied by the presence of fire."

1.7.2 "Only if"

It's also possible (but much less common) to label the *conclusion* instead of the hypothesis. Often the conclusion is labeled as something that necessarily must be true, often that that it's the *only* way that the hypothesis could be true. Consider the following sentence:

Example 1.5. "Our plan to rob the bank will succeed only if you turn off the security cameras."

S: "Our plan to rob the bank will succeed."
O: "You turn off the security cameras."

You might be inclined to translate this as $O \rightarrow S$, but if that was the correct translation, then that would be saying that if you turn off the cameras, then the plan is guaranteed to succeed. But that's not necessarily true at all. There are all sorts of other ways the plan could go wrong.

In fact, the best way to think about “only if” is to think of it as identifying a *conclusion*, not a premise. So if the original sentence “Our plan to rob the bank will succeed only if you turn off the security cameras” is true, then you know that if the plan succeeds, then it must be because you turned off the cameras. Knowing the success of the plan leads you to knowing that the cameras must’ve been turned off.

This sort of construction is confusing because the conclusion (turning off the cameras) happened *before* the premise (the successful). In fact, you might even be inclined to say that the premise was *caused* by the conclusion. Although, if we want to be technical about it, we should really say that the conclusion caused the premise *to be possible*. Even though turning off the cameras is what made it possible for the robbery to succeed, the conditional works in the opposite direction, and the correct translation is $S \to O$.

This is not an unusual example. Most of the time, when you use “only” to modify a premise tag like “if” or “when”, it will be a situation where the conclusion being true is one of the things that makes *causes* it to be possible for the premise to be true. But the direction of the conditional always works in the opposite direction from this

My general rule of thumb:

> Putting the word “only” in front of a premise tag turns it into a conclusion tag.

Example 1.6. “The patient has leg pain only when sitting down.”

P: “The patient has leg pain.”

S: “The patient is sitting down.”

This doesn’t mean that patient feels pain *every time* they sit down. But it does mean that every time they feel pain, they are sitting down.

Translation: $P \to S$

In the above example, the sentence tells us that if the patient’s leg hurts, then they must be sitting down ($P \to S$), but when the patient is sitting down, it does not tell us whether or not they always feel pain. It might be that when sitting, they only sometimes feel pain. Or it might be true that *every* time they sit, their leg hurts. In other words, $S \to P$ might *also* be true, or it might not. This particular sentence just doesn’t give us that information.

The rule of thumb I gave you above is pretty reliable. In this class, you should always follow this rule. I always try to use example sentences that are as clear-cut as possible. However, if you grab a random example of “only if” from the real world, it is very likely going to be saddled with a lot of confusing context. It’s hard to separate the specific thing that a sentence is *communicating* from the stuff that we happen to know from context.

Here’s an example of a sentence that comes with some confusing contextual baggage:

Example 1.7. “We will contact you only in the case that there is an issue with the payment.”

C: “We will contact you.”
I: “There is an issue with the payment.”
Translation: $C \to I$

To some people, the above translation might seem to be only partially correct. There's no doubt that it's at least part of what the original sentence is saying: if you are contacted, you know that it must be because there was an issue with the payment. But given what we know from our experiences buying things online, we can make a good guess that the opposite direction ($I \to C$) is also likely true. In other words, we can be pretty sure that *every* time there's an issue with the payment, you will be contacted.

So does that mean that we should really be translating this as $(C \to I) \wedge (I \to C)$ based on context?[8] I don't think so. I think this is a case where we know that $I \to C$ is true based on our knowledge of how online systems work, but it's not part of what the sentence “We will contact you only in the case that there is an issue with the payment” is actually communicating. The sentence is just telling us this new piece of information: if we contact you, you'll know that it has to do with the payment. Everything else is context; it's not part of the sentence itself.

1.7.3 Causality

People often confuse conditional implication with causality. Propositional logic has nothing to do with cause and effect, so don't fall into this trap! When we make a claim of the form $p \to q$, we're *not* saying that p *causes* q to be true. Instead, we're saying that in those situations where we know that p is true, we can conclude that q must also be true.

The reason *why* we know that a particular implication is true might have something to do with cause and effect, but that's not part of what → means. So for example, the reason why a sentence like “If you open the door, the cat will try to get out,” is true might be because opening the door “causes” the cat to try to leave, but we only know this because of our understandings about cats and doors. The sentence itself just tells you that any time the door is open, you can be sure that the cat will try to escape. In many other situations, cause and effect has nothing to do with it at all, such as with the zip code example from earlier. And even when cause and effect is involved, it's the *conclusion* that is the cause and the *premise* that is the effect. In the example “If there is smoke, there is fire,” it's the fire that's causing the smoke, but the *implication* states that seeing smoke is *enough to conclude* that there must be a fire.

Propositional logic just isn't powerful enough to describe the convoluted and confusing concepts of cause and effect, so if a sentence has words like “because”, “since”, “therefore”, “hence”, or ”so”, we will not be able to fully translate those sentences into propositional logic. In fact, the closest we can get is actually to use *conjunction* and not implication.

[8] This is the same thing as $C \leftrightarrow I$, but we haven't gotten to ↔ yet.

For example, a sentence like "I got wet because it rained," does not mean the same thing as "If it rained, then I got wet." Using "if" here implies that the speaker doesn't even know whether or it rained, but the sentence "I got wet because it rained," makes it absolutely clear that it did in fact rain, *and* that the speaker got wet. It *also* communicates that the *cause* of the speaker wetness was the rain, but that part of the sentence can't be captured by propositional logic.

I will not ask you to translate sentences involving these kinds of words into propositional logic because propositional logic isn't a good fit for them. And you should not use such words in your translations *from* propositional logic.

1.8 Biconditional Implication

Of course, we sometimes want to communicate that the implication *does* go both ways. Suppose we are running research subjects through a maze. We want to make sure that the subjects are rewarded when they finish the maze, so we say "The rat gets a reward if they complete the maze." But we also want to make sure that there's no other way for them to get the reward, so we also say "The rat gets a reward only if they complete the maze." We could simply put these two requirements together in the sentence "The rat gets a reward if they complete the maze, and the rat gets a reward only if they complete the maze." Of course, that sentence has a lot of repeated words in it, and humans love to drop repeated parts of sentences, so in English, we tend to shorten this to "The rat gets a reward if and only if they complete the maze."

This kind of construction is so common that we tend to think of the phrase "if and only if" as a single unit, important enough to get its own name and symbol. It's even common to abbreviate the phrase "if and only if" as just "iff". We call this kind of connection **biconditional implication** or **bi-implication**.

For obvious reasons, the most common symbol for the biconditional is a two-headed arrow $\leftrightarrow$ (LaTeX: `\leftrightarrow`, Unicode: `U+2194`, HTML: `&harr;`), which is usually read out loud as "if and only if". Some older books use a double-lined arrow $\Leftrightarrow$ or the equivalence symbol $\equiv$. We'll be using the $\equiv$ symbol for a different (but related) purpose later on (see the section on logical equivalence).

Let's take a look at the truth table for logical equivalence:

p	q	$p \leftrightarrow q.$
T	T	T
T	F	F
F	T	F
F	F	T

One way to summarize this table is to point out that $p \leftrightarrow q$ is true when p and q have the same truth value (when they're both true or both false). So we often think of the biconditional as communicating that the two parts are somehow *equivalent* to each other. So you'll often see the phrase "if and only if" (or its abbreviation "iff") used in definitions. For example: "A number is even *if and only if* it can be written as $2n$ for some integer n.

1.8.1 “Necessary” and “Sufficient” Conditions

In technical circles, you’ll occasionally hear people say one piece of information is a “necessary condition” or a “sufficient condition” for another piece of information. For example, someone might say “Matching fingerprints are a sufficient condition to establish the presence of the suspect in the room,” or “Being plugged in is a necessary condition for the television to be on.” Let’s look at these two examples to get a feel for what’s going on.

Example 1.8. “Matching fingerprints are a sufficient condition to establish the presence of the suspect in the room.”
F: “The fingerprints match.”
R: “The suspect was in the room.”
Translation: $F \to R$

Saying that “matching fingerprints” is a “sufficient condition” tells us that knowing that the fingerprints match is *enough information* to establish some other fact. In this case, the sentence is telling us that if we know that the fingerprints match, then we can conclude that the suspect was in the room. A “sufficient condition” is really just another way of saying “condition” or “premise”. When you see “sufficient condition” by itself, it’s usually there to emphasize the fact that the condition is enough evidence *all by itself* to prove some conclusion without any other evidence needed.

As a side note, notice that this kind of construction requires the two propositions to be expressed by noun phrases (“matching fingerprints” and “the presence of the suspect in the room”), but when we do the translation, it’s clearer to phrase them as actual sentences (“The fingerprints match,” and “The suspect was in the room.”)

Example 1.9. “Being plugged in is a necessary condition for the device to be on.”
P: “The device is plugged in.”
O: “The device is on.”
Translation: $O \to P$

On the other hand, a “necessary condition” is a different thing entirely. By telling us that being plugged in is a “necessary condition”, this sentence is telling us that the *only* way the device can be on is if it is plugged in.[9] In other words, if the device is on, then we can conclude that it is plugged in ($O \to P$). A necessary condition is often only *one* of the things required to establish some fact as true, and so we don’t automatically know $P \to O$. If the device is plugged in, we can’t tell whether or it is on. Maybe the device has an on-switch (like a television) that also needs to be flipped? Or maybe it doesn’t (like a set of Christmas lights)! The sentence doesn’t give us enough information.

Now typically, you don’t encounter “necessary condition” or “sufficient condition” by themselves. You are far more likely to encounter both together, as in:

[9] This should remind you of “only if”, from section 1.7.2.

Example 1.10. "Being plugged in is a necessary and sufficient condition for the Christmas lights to be on."

P: "The Christmas lights are plugged in."

O: "The Christmas lights are on."

Translation: $O \leftrightarrow P$

Since being plugged in is a *sufficient* condition, then we know that as soon as we plug in the lights, they will be on ($P \to O$). (So we know there isn't any other required condition, like a light switch.) Since being plugged in is also a *necessary* condition, then we know the converse as well: if the lights are on, it must be because they are plugged in ($O \to P$). (There isn't a battery or some other power source.)

The other common place where you will encounter "necessary" and "sufficient" conditions is when someone wants to explicitly draw a contrast between them, as in:

Example 1.11. "Being plugged in is a necessary, but not sufficient condition for the television to be on."

P: "The television is plugged in."

O: "The television is on."

Unfortunately, we can only partially capture this in propositional logic. Since being plugged in is a necessary condition for the TV to be on, we can capture this part of the translation with $O \to P$. Now since we know that it's *not* a sufficient condition, we know that it's *possible* to have the TV plugged in at the same time that the TV is off. Unfortunately, propositional logic isn't powerful enough to talk about things being "possible". Propositional logic can only talk about what is true and what isn't true in one specific situation.

If we want to talk about what's "possible", we'll need to expand our notion to be able to talk about more than one situation at a time. When we get to first-order logic and we introduce the ideas of universal and existential quantifiers, we'll be able to handle this kind of situation properly. But for now, we just have to remember that propositional logic isn't powerful enough to capture the idea of an implication *not* being true.

1.9 Truth Assignments and Satisfaction

Okay, enough about translation for now. What can we say about a propositional formula by itself, without any particular meaning associated to the variables? The first thing to understand is that even though a formula represents a proposition (which is something that can be either true or false), it does not make sense to talk about whether a propositional formula is true or false by itself. It's a lot like in elementary algebra, where it doesn't make sense to ask "Is $x+y \leq 5$ true?" You *could* ask "If we set x to 4 and y to 3, is $x + y \leq 5$ true?" Or, to put it another way, "Does the assignment $(x = 4, y = 3)$ make the statement $x + y \leq 5$ true?" Or better yet: "Does the assignment $(x = 4, y = 3)$ **satisfy** the statement $x + y \leq 5$?"

We can do exactly the same thing for formulas of propositional logic. The only difference is that instead of algebraic variables standing in for numbers, we have propositional variables standing in for atomic propositions. Each propositional variable can take on either the value of "true" or the value of "false". An assignment of truth values to each propositional variable is called a **truth assignment**, or often just an **assignment**.

WARNING! "Truth" sounds a lot like "true", so be careful. "Truth", in this context, means "whether or not something is true". So a *truth* assignment for a variable can be either "true" or "false". I won't ever use the phrases "true assignment" or "false assignment", and you shouldn't either. Why not? Because it's not clear whether "true assignment" means "an assignment that assigns 'true' to the atomic variable(s)" or "an assignment that makes the entire formula 'true' ". If you want to talk about an assignment that makes the formula "true", it's much better to call it a "satisfying assignment".

To sum up: While we can't talk about whether the formula $P \to Q$ is true or false, we can talk about whether a particular truth assignment, maybe $(P = \mathrm{T}, Q = \mathrm{F})$, satisfies $P \to Q$. What do you think? Does it?

Example 1.12. Here are a few more practice questions for you to consider:

1. Does $(P = \mathrm{F}, Q = \mathrm{T})$ satisfy $P \vee \neg Q$?

2. Does $(P = \mathrm{T}, Q = \mathrm{T})$ satisfy $(P \wedge \neg P) \to Q$?

3. Is there a truth assignment that satisfies $P \leftrightarrow (\neg P \wedge Q)$?

4. Can $P \wedge \neg P$ be satisfied?

The first two questions can be answered by plugging in the corresponding truth values and doing a basic calculation. You could represent this using a single row in a truth table:

P	Q	$\neg Q$	$P \vee \neg Q$
F	T	T	T

So yes, $(P = \mathrm{F}, Q = \mathrm{T})$ does satisfy $P \vee \neg Q$.

Or you could treat it like an algebraic calculation, "plugging in" T or F for the atomic variables and computing from there:

$$\begin{aligned}(P \wedge \neg P) \to Q &= (\mathrm{T} \wedge \neg \mathrm{T}) \to \mathrm{T} \\ &= (\mathrm{T} \wedge \mathrm{F}) \to \mathrm{T} \\ &= \mathrm{F} \to \mathrm{T} \\ &= \mathrm{T}\end{aligned}$$

Again, we get a "yes" answer; $(P = \mathrm{T}, Q = \mathrm{T})$ satisfies $(P \wedge \neg P) \to Q$.

Now let's take a look at question number 3. This simplest solution is just to try all the truth assignments one at a time until we find one that works. When we're interested in looking at many truth assignments at once, it's useful to build a table, like the ones we built when we were defining the connectives. It allows us to see at a glance all the satisfying and non-satisfying truth assignments the formula may have.

P	Q	$\neg P$	$\neg P \wedge Q$	$P \leftrightarrow (\neg P \wedge Q)$
T	T	F	F	F
T	F	F	F	F
F	T	T	T	F
F	F	T	F	T

Some students end up treating truth tables as if they were an important, fundamental feature of propositional logic, but they really aren't. The important, fundamental feature of propositional logic is the *truth assignment.* A truth table is nothing more than a way to keep track of all the truth assignments in a systematic way.

Each row represents a truth assignment, as defined by the first few columns in the table. In any given column, a value of T indicates that the truth assignment for that row *satisfies* the formula at the top of the column. And a F entry indicates that the corresponding truth assignment does *not* satisfy the formula.

In this case, we're only interested in that last column, the one for the full formula $P \leftrightarrow (\neg P \wedge Q)$. Since there is a row that has T (the last one), we know that there is an assignment that satisfies the formula (specifically $(P = \mathrm{F}, Q = \mathrm{F})$.

Once you get used to filling out tables, you may find that you want to skip columns to save time and effort. I recommend against skipping columns because it makes errors much more common. It might seem like that $\neg P$ column is so obvious as to not be needed, and you're right in that students rarely make mistakes in filling out the $\neg P$ column, but I've noticed that when students *skip* that column, they are much more likely to make a mistake on the *next* column.

A bit of general advice: If each step you do seems so simple as to be boring, that's a *good* thing. The whole idea of the process is to break down complex procedures into simple, boring steps.

Finally, let's address the last question "Can $P \wedge \neg P$ be satisfied?" Perhaps you have a good idea about the answer just by looking at the formula, but even if you don't, we can build the (very small) truth table for that formula just to be sure:

P	$\neg P$	$P \wedge \neg P$
T	F	F
F	T	F

So no, there is no assignment that satisfies $P \wedge \neg P$.

What is the significance of a formula that has no satisfying truth assignments? Suppose we gave a meaning to P, just to bring things back to where we're comfortable thinking about them. Perhaps P means "the TV is on." Then $P \wedge \neg P$ would mean "The TV is on, and the TV is not on," which we can all see is false, even without knowing anything about the TV in question. If we change the meaning of P to, say "12 is an even number," then $P \wedge \neg P$ would mean "12 is an even number, and 12 is not an even number," another false statement. In fact, *no matter what meaning we assign to P, the sentence will always be false.*

Unlike ordinary formulas, the truth of the formula $P \land \neg P$ does not depend on the meaning of its atomic variables. The logical structure alone is enough to tell us that the proposition can never be satisfied. The two conjuncts P and $\neg P$ contradict each other regardless of whether P is true or not. Such a formula (one with no satisfying assignments) is called a **contradiction**.

Definition 1.2. A formula of propositional logic is a **contradiction** if and only if no truth assignment satisfies it.

This property of being a contradiction is an example of what we call a **universal** property, meaning that it's a claim about *everything* in some particular "universe". In this case, that "universe" is the set of all truth assignments. Justifying a universal claim is usually a fairly involved process because it requires verifying that a claim is true in lots of different situations (in this case, we need the formula to not be satisfied in all of the different truth assignments).

*Dis*proving a universal claim is much simpler, because then you just have to find a single counterexample. When a formula is *not* a contradiction, that means that there is *at least one* truth assignment that satisfies it. In other words, a formula is not a contradiction if it is *possible* to satisfy the formula.

Definition 1.3. A formula of propositional logic is **satisfiable** if and only if there is at least one truth assignment that satisfies it.

Satisfiability is an example of what we call an **existential** property, meaning that it's a claim that only depends on the existence *at least one* example meeting the needed requirements. Existential properties are easy to prove (just come up with the example), but they are much more difficult to disprove, as that requires demonstrating that *no* possibility works as an example.

Existential and universal properties always come in pairs. The negation of every existential claim is a universal claim. (If there does not exist an example with a given property, then everything in the universe fails to have that property.) The negation of every universal claim is an existential claim. (If it's not true that everything has a particular property, then there must be at least one counterexample that fails to have that property.) When we get to first-order logic, we'll study universal and existential claims directly, but it's good to start thinking about which properties are universal and which are existential in a general sense.

With this in mind, here are a couple important guidelines for solving problems about satisfiability in this class. **To prove that a formula is satisfiable, you must give an example of a satisfying truth assignment.** Conversely, **to prove using tables that a formula is *not* satisfiable (i.e., that it is a contradiction), you must give an entire table, with false in every row.**

Question. Are there formulas with no *non*-satisfying truth assignments, ones that are satisfied by every truth assignment? Can you think of an example?

Indeed, there are! One example would be $P \to P$. If you can't see why, you should build the truth table and verify it. In fact, you should probably build the truth table anyway; it'll be good practice. And it's a short table too.

Formulas like $P \to P$ are true no matter what truth value you assign to the propositional variables. If the variables were to be given a meaning in English, the resulting statement would be true by virtue of logic alone. ("If the TV is on, then the TV is on.") Despite always being true, claims like this are actually pretty useless. They are so obviously true that there's rarely any reason for them to be spoken, except when someone is trying to be a smart-aleck. We call these formulas **tautologies**.

Definition 1.4. A formula of propositional logic is a **tautology** if and only if it is satisfied by *every* truth assignment.

Note that being a tautology is a *universal* property about truth assignments, so proving that a formula is a tautology requires you go through every single truth assignment. If you're using a table, this means that you need to give the entire table. **To prove that a formula is a tautology using tables, you must give *every* row of the table.** That can be a lot of work if there are a lot of atomic variables!

On the other hand, disproving such a claim is pretty easy. Since being a tautology is a universal property, then the claim that a formula is *not* a tautology is an *existential* claim. This means you only have to find *one* assignment that fails to satisfy the formula to prove that it is not a tautology. You might *find* this truth assignment by building a table, but the ultimate proof is just going to be the one non-satisfying truth assignment. **To prove that a formula is *not* a tautology, you must always give a truth assignment that does not satisfy the formula.** I'm going to hold you to this rule even if you've already given the entire table. I want to make sure you can read the corresponding truth assignment from the truth table, so you have to explicitly give the truth assignment too. The table is just the work you did to figure out what assignment will work.

Both tautologies and contradictions are kind of pointless, from the viewpoint of translations. Most normal formulas are neither contradictions nor tautologies. The truth values of such formulas are dependent on whether their atomic variables are true or false. Or, to use a slightly fancier word for "dependent", we could say that they are *contingent* upon the truth values of their atomic variables.

Definition 1.5. A formula of propositional logic that has at least one satisfying assignment and at least one non-satisfying assignment is called a **contingency**.

The property of being a contingency is an existential claim. Actually, it's the conjunction of *two* existential claims. **To prove that a formula is a contingency, you must give *two* truth assignments: one satisfying, and one not satisfying.**

And of course, the negation is a universal claim. **To prove that a formula is *not* a contingency using tables, you must give an entire table, showing either that every assignment satisfies the formula or no assignment does.**

Note that every propositional formula falls into exactly one of the above categories: tautology, contradiction, or contingency. If you're looking for a little more practice, you could take all of the formulas we discussed above and figure out which category they all fall into.

Let's do a few example problems just to make sure you understand all these definitions, and how much justification you will be expected to give.

Example 1.13. Is $P \to (Q \to P)$ satisfiable?

To answer this question, we need to know if there is a truth assignment that satisfies the formula. One way would be to just start thinking up random truth assignments and trying them out. That can work, especially if you have a good idea where to start, but a truth table has several advantages. It makes it easier to keep track of which assignments have already been tried, it helps us track down errors more easily, it makes the process of coming up with new truth assignments to try super simple, and in case it turns out that there are no satisfying assignments, the table itself can serve as proof of that fact.

P	Q	$Q \to P$	$P \to (Q \to P)$
T	T	T	T
T	F		
F	T		
F	F		

Why did I stop there? Because the question just wanted to know if it *could* be satisfied. We found a truth assignment that does the job, namely $(P = \text{T}, Q = \text{T})$. Maybe there are others, but right now, we don't care about them.

Solution. Yes, because $(P = \text{T}, Q = \text{T})$ satisfies the formula.

In this case, the table was just a tool to find this truth assignment. The truth assignment is all we need to *prove* that the formula is satisfiable. Remember that whenever you claim that a formula is satisfiable, you should provide a truth assignment that proves this.

Example 1.14. Is $P \to (Q \to P)$ a tautology?

If we think the answer is going to be "no", we could just guess truth assignments until we find one that doesn't satisfy the formula. But a table is a more reliable strategy. Even if you've got a good guess for a truth assignment that doesn't satisfy the formula, you can still put it in a table, just start with the row that you think is most likely to give you a "false" result. Of course, if you think the formula *is* going to be a tautology, then we're going to need the entire table anyway.

P	Q	$Q \to P$	$P \to (Q \to P)$
T	T	T	T
T	F	T	T
F	T	F	T
F	F	T	T

Remember that each row is checking a particular truth assignment. If we'd found even a single non-satisfying assignment (we'd know because there'd be an F in the last column), then that would mean the formula was *not* a tautology. If that was the case, we could stop as soon as we found the assignment that didn't satisfy the formula, say the answer was "No", and give that non-satisfying truth assignment as a counterexample. Remember that whenever you claim that a formula is *not* a tautology, you should give a truth example that proves this.

Of course, that's not the case here. This formula *is* a tautology because *every* assignment satisfies the formula. Of course to prove this, we need to make sure that we prove it for all the assignments (meaning every row of the table, if we're using tables). Remember that to prove that a formula is a tautology using tables, you must give *every* row of the table, so your answer might look like this:

Solution. Yes.

P	Q	$Q \to P$	$P \to (Q \to P)$
T	T	T	T
T	F	T	T
F	T	F	T
F	F	T	T

Question. How would you prove that a formula is a *contingency*? How would you prove that a formula is *not* a contingency? How about contradictions?

formula	to prove	to disprove
satisfiable	1 satisfying assig.	no assig. satisfies
tautology	all assig. satisfy	1 non-satisfying assig.
contradiction	no assig. satisfy	1 satisfying assig.
contingency	1 satisfying & 1 non-satisfying	all satisfy or none satisfy

Question. If you had been translating an English proposition and the translation turned out to be $P \to (Q \to P)$, what would that tell you about the original sentence?

Answer. It would mean that no matter what was true about the individual parts of the sentence, the whole thing would always be true. But it's only true for logical reasons, meaning it's actually kind of a stupid thing to say. The answer isn't dependent on reality, so what's the point in even mentioning it?

Question. Since $P \to (Q \to P)$ is a tautology, what (if anything) does that tell us about the formula $\neg\big(P \to (Q \to P)\big)$?

Answer. Well, the negation of a formula is true when (and only when) the original formula is false. (And so it's also true that it's false when and only when the original formula is true.) So if the original formula is always true, then the negation has to be always false. So we know that $\neg\big(P \to (Q \to P)\big)$ must be a *contradiction.*

Warning! When we're talking about "*the* negation of a tautology", this is not the same thing as talking about "*a* formula that is not a tautology". When we say that F is "*the* negation of a tautology", we mean that there's a specific tautological formula (call it G) and that F is just what you get when you put $\neg$ in front of G (in other words, F is the same thing as $\neg G$). When we say that F is "*a* formula that is not a tautology", we just mean that F is a formula that happens not to be a tautology. There's no other formula in mind, and there's not necessarily any $\neg$ around, so you shouldn't be thinking about flipping all the truth values around. Which brings us to another interesting question:

Question. What can you say about a formula that is not a tautology? Does it have to be a contradiction? *Can* it be a contradiction?

1.10 Pairs of Formulas and Logical Equivalence

So far, we've been dealing with what you can say about individual propositional formulas on their own. And, unless they're really big, complicated formulas, there's not a whole lot you can say about them. You can talk about which assignments satisfy them (more specifically, whether any assignments satisfy them, whether all assignments satisfy them (or none do), or whether we have a mix of satisfying and non-satisfying assignments), but that's about it. Things become more interesting when we talk about how different formulas relate to each other. (Of course, in the long run, everything always boils down to which assignments satisfy which formulas.)

Consider the pair of formulas $A \vee B$ and $B \vee A$. Would you say these are "different" formulas? Sure, on the surface, they look a little different, but we think of them as having the same meaning. Maybe there's some difference in whether we want to emphasize the A or the B, but as far as the logic is concerned, they're effectively the same. They're different formulas, but when it comes to the important stuff, they act the same way. If either A or B is true, then both $A \vee B$ and $B \vee A$ are true, and both formulas are false for the truth assignment $(A = \mathrm{F}, B = \mathrm{F})$. In other words, they are satisfied by exactly the same truth assignments (and as a result, they *fail* to be satisfied by exactly the same assignments). As far as the logic is concerned, wherever we use one of them, we could just as easily use the other one. So we say they are **logically equivalent**. More formally:

Definition 1.6. A pair of formulas are **logically equivalent** (sometimes we just say **equivalent**) if and only if every assignment that satisfies one of them also satisfies the other. When a pair of formulas p and q are logically equivalent, we write $p \equiv q$.

Note that this is a *universal* property, so **if you want to prove using tables that a pair of formulas are logically equivalent, you must give an entire table, with identical truth values for the two formulas.** And as a universal property, it's negation is an *existential* property. So, **if you claim**

a pair of formulas are *not* equivalent, you must prove this by giving an assignment that satisfies one formula but not the other.

That triple equal sign $\equiv$ is meant to remind you of an equal sign. In math, when we write something like $\frac{1}{2} = 0.5$, we're really saying that anytime you use $\frac{1}{2}$ for anything in math, you could use 0.5 and get the exact same results. The same thing goes for equivalence and logic. Any place you see $A \vee B$, you could replace it with $B \vee A$, and you'd get the same result.

For example, if you were trying to decide whether $(A \vee B) \to \neg A$ was a contradiction or not, instead of looking at the table for $(A \vee B) \to \neg A$, you could look at the table for $(B \vee A) \to \neg A$. In this case, I don't see *why* you would want to do that (maybe you already built the table for $(B \vee A) \to \neg A$ in a previous problem?), but if you did, you would get the same answer.

If you're having trouble deciding between two potential translations of an English sentence into propositional logic, and if those two potential translations are logically equivalent, then it actually doesn't matter which one you use. For example, some people (including me) might translate a sentence of the form "P, unless Q," as $P \vee Q$, while other people might translate it as $\neg P \to Q$, and others might translate it as $\neg Q \to P$. Who is correct? We all are! All of those formulas are logically equivalent (go ahead and check the truth tables if you don't believe me), so if any one of them is an acceptable translation, then they *all* are acceptable.

If the formulas are simple, it's usually pretty obvious whether two formulas are logically equivalent, but there are some surprises. Certainly $(A \wedge B) \wedge C \equiv A \wedge (B \wedge C)$ is true.[10] But what about if you replace $\wedge$ with $\to$?

Example 1.15. $(A \to B) \to C \equiv A \to (B \to C)$?

We have to decide whether or not these two formulas are logically equivalent. If we think the answer is "no" (i.e., if they're not equivalent), then all we need to do is find a single truth assignment that satisfies one of the formulas but not the other. We could just guess truth assignments until we find one that works, but that won't work if the answer turns out to be "yes". If they *are* equivalent, we need to show that for every assignment, the two formulas have the same truth value. Again, our best tool so far is the truth table.

A	B	C	$A \to B$	$(A \to B) \to C$	$B \to C$	$A \to (B \to C)$
T	T	T	T	$\boxed{\text{T}}$	T	$\boxed{\text{T}}$

So this first assignment ($A = \text{T}, B = \text{T}, C = \text{T}$) makes both formulas true. Does that tell us anything? Not really. If it had made one formula false and the other true, then we'd know the answer was no. As it is, we need to keep going.

[10] By the way, this is why you're allowed to write $A \wedge B \wedge C$ without parentheses. No matter where the parentheses go, you end up with equivalent formulas. It's also why you should *not* write $A \to B \to C$.

A	B	C	$A \to B$	$(A \to B) \to C$	$B \to C$	$A \to (B \to C)$
T	T	T	T	**T**	T	**T**
T	T	F	T	**F**	F	**F**
T	F	T	F	**T**	T	**T**
T	F	F	F	**T**	T	**T**
F	T	T	T	**T**	T	**T**
F	T	F	T	$\boxed{\mathbf{F}}$	F	$\boxed{\mathbf{T}}$
F	F	T				
F	F	F				

The second row has both formulas false, so the assignment ($A = \text{T}, B = \text{T}, C = \text{F}$) doesn't satisfy either formula. That's still okay as far as equivalence is concerned, so we keep going. Eventually we find a row where one is false, and the other is true. That's the key to our answer! Make sure you include that truth assignment as part of your answer. So if you guessed this truth assignment right off the bat, you'd be in luck and you wouldn't need to fill out the whole table. Of course, if you claim a pair of formulas *are* equivalent, then you need to show that *all* assignments result in the same truth value for both formulas. If you're doing it with tables, that means you need the whole table.

Solution. No. ($A = \text{F}, B = \text{T}, C = \text{F}$) satisfies $A \to (B \to C)$ but not $(A \to B) \to C$.

It's possible to reformulate this concept of logical equivalence in terms of a single formula.

Fact. The formulas p and q are logically equivalent if and only if the formula $p \leftrightarrow q$ is a tautology.

Think about this fact and make sure you understand why it is true.

1.11 Sets of Formulas and Consistency

Consider the two formulas $A \wedge \neg B$ and $A \leftrightarrow B$. Both are satisfiable by themselves, but they can't possibly be true at the same time because they contradict each other. To be more precise: the formulas can't be satisfied by the same truth assignment. Whenever a set of formulas (in this case, just two of them) can't be made true by the same truth assignment, we say that the set of formulas is **inconsistent**. As you'd expect, there's also a definition for **consistency**.

Definition 1.7. A set of formulas is **consistent** if and only if there exists at least one truth assignment that satisfies all of the formulas in the set. A set of formulas that is not consistent is said to be **inconsistent**.

We use the word "consistent" here in the same way that we might say that a work of fiction is internally consistent. We don't mean that the events in the work are actually true, merely that they don't contradict each other.

"Set" has a technical meaning, which we will discuss in detail later, but for now, it's enough to know that by a set of formulas, we just mean any collection of formulas. We'll indicate that we're dealing with a set of formulas by using curly braces to surround them and commas to separate them, like this: $\{A \wedge \neg B, A \leftrightarrow B\}$. As hinted at earlier, this particular set is inconsistent. How can we tell? Let's start by looking at the truth table for the formulas.

A	B	$A \wedge \neg B$	$A \leftrightarrow B$
T	T	F	T
T	F	T	F
F	T	F	F
F	F	F	T

In order for this set to be consistent, there would have to be a single truth assignment that makes all of the formulas in the set true. In terms of truth tables, this means there has to be a row in which the entries for the formulas in the set are all "true". Since there is no such row, the set must be inconsistent. To prove this, we have to somehow indicate that *no* truth assignment satisfies all of the formulas. One way to do this is to give the entire table. We'll see other ways later.

Let's look at an example that does turn out to be consistent. Consider the set $\{A \vee B, B \rightarrow A, \neg B\}$.

A	B	$A \vee B$	$B \rightarrow A$	$\neg B$
T	T	T	T	F
T	F	T	T	T

Let's stop right there. We've found an assignment that satisfies all three formulas, so there's no need to finish the table. The proof we need comes from that second row. Anyone who wants to know that the set is consistent doesn't care about the rest of the table. All they care about is that one truth assignment, so for a justification, we give the truth assignment $(A = \mathrm{T}, B = \mathrm{F})$.

Note that consistency is an existential property. **To prove that a set of formulas is consistent, you must give a truth assignment that satisfies all the formulas in the set.** And as you've probably guessed by now, *in*consistency is a universal property. **To prove using tables that a set of formulas is inconsistent, you must give an entire table, where none of the rows has true for all the formulas in the set.**

Consistency and satisfiability are very closely related. They're both existential claims about satisfying truth assignments, but they talk about different kinds of things. Always remember that a formula has to be consistent "with" other formulas. So it makes sense to say "the set is consistent" because that's the same as saying "the formulas in the set are consistent with each other". But it doesn't make sense to just say "the formula is consistent." Similarly, it doesn't make sense to talk about a set of formulas as being "satisfiable". If you do say that, we don't know if you mean that each formula is satisfiable (which doesn't tell us if the set is consistent or not) or if you mean that they can all satisfied by the same formula (which means the set is consistent).

As with equivalence, it's possible to reformulate the concept of consistency in terms of a single formula, although since this is an existential claim, it won't be about tautologicity, but about satisfiability.

Fact. The set of formulas $\{p_1, p_2, \ldots, p_n\}$ is consistent if and only if the formula $p_1 \wedge p_2 \wedge \cdots \wedge p_n$ is satisfiable.

Again, I'd like you to think about this fact and make sure you understand why it is true.

1.12 Arguments and Validity

All this talk of how to satisfy formulas is great and all, but the core of logic circles around the concept of an *argument*. This is when we try to argue that some claim must be a logical conclusion of some other claim (or set of claims). There's a whole lot that goes into argumentation, but since this is a math and logic course, we're going to ignore aspects such as emotional appeals, fact-checking, and shouting. Instead we'll focus on the logical aspects of an argument. Take the following example of an argument using English.

- Either the number is even, or it is odd.
- The number is not odd.

Therefore, the number is even.

You don't even have to know what the words "number", "even", and "odd" mean in order to follow this argument. If you buy into the two premises ("either the number is even, or it is odd" and "the number is not odd"), then you must also believe the conclusion ("the number is even"), unless you refuse to believe in even the most basic logic.

Definition 1.8. In symbolic logic, an **argument** is a set of formulas (called the **premises**) and a special formula (called the **conclusion**).

That's all there is to being an argument. Any set of formulas can be a set of premises and any formula can be a conclusion. Of course, if the premises have nothing to do with the conclusion, then the argument is probably a very silly one. Mostly, we are interested in arguments where the conclusion is a logical consequence of the premises. We use the word "valid" to describe such an argument.

Definition 1.9. An argument is **valid** if and only if every truth assignment that satisfies all the premises also satisfies the conclusion.

This definition is deceptively complex. I find it much easier to think of validity in terms of its negation. If an argument is not valid, it is (surprise, surprise) said to be **invalid**. Here's how I like to think about invalidity:

Definition 1.10. An argument is **invalid** if and only if there is at least one truth assignment which satisfies all the premises, but *not* the conclusion.

Using this definition, it's easy to look through a table and see if there are any rows with "true" for every premise and "false" for the conclusion. If you find even one such row, then the argument is invalid. If you can't find any, then it is valid.

Using the other definition is a little harder to write down in words. You go through the rows of the table. If you find a row with "true" for all the premises, and "true" for the conclusion, then so far so good. (You're not done because you still have to check *all* the rows!) If any of the premises are "false" in this row, then that is also "so far so good". You only get to say that the argument is valid if you get to the end and *every* row falls into one of the two above cases.

In actuality, both "techniques" are really the same technique. You're not doing anything different in either of them. They're just two different ways of thinking about the same method.

Let's apply this method to the example from above. Translating our English example above to logic:

- E: The number is even.
- O: The number is odd.

$$\begin{array}{l} E \lor O \\ \neg O \\ \hline E \end{array}$$

Notice that the word "therefore" does not get translated as a logical connective (like $\rightarrow$). It is not connecting two parts of a single proposition, so it shouldn't be thought of as a connective. The purpose of the word "therefore" is to indicate that the proposition that follows is the conclusion. When we translate the argument, we also have to indicate that E is the conclusion. We did this by putting a horizontal line above it, separating it from the premises. *This line is not optional.* Without it, you've just got a list of formulas, without any sort of connection between them.

Let's look at the truth table for these formulas, checking to see if this is a valid argument.

	E	O	$E \lor O$	$\neg O$	E
	T	T	T	F	T
$\Longrightarrow$	T	F	T	T	T
	F	T	T	F	F
	F	F	F	T	F

Looking at the truth values for the premises, we see that we only really care about the truth value of the conclusion for one assignment (the one in the second row). And this one truth assignment that makes all the premises true also makes the conclusion true. Validity is a claim about all truth assignments, so we need the entire table to justify our answer. Don't just give that one truth assignment! One truth assignment is never enough to prove validity. In this case, there only happened to be one assignment that made all the premises true, so there's only one row in which we care about the last column. But that's only because there was only one such row.

Let's look at an argument where there is more than one way to satisfy all the premises.

$$\begin{array}{l} A \to \neg B \\ \neg B \\ \hline A \end{array}$$

	A	B	$A \to \neg B$	$\neg B$	A
	T	T	F	F	T
$\Longrightarrow$	T	F	T	T	T
	F	T	T	F	F
$\Longrightarrow$	F	F	T	T	F

In this case, we need to look at the conclusion for two truth assignments (in the second and fourth rows). The second row's assignment satisfies the conclusion... so far, so good. But the fourth row's assignment makes the conclusion false. This is enough to tell us that the argument cannot be valid. It's possible for the premises to be true and the conclusion to be false. In this case, we had to fill out the entire table to find this assignment, but the only evidence we need is this truth assignment ($A = \text{F}, B = \text{F}$) that satisfies the premises but not the conclusion. A valid argument is similar to a tautological formula because proving either requires a lot of evidence, but disproving either can be done with a single counterexample.

If you've been paying attention to the last few sections, you've probably guessed that there's a way to reformulate the concept of consistency in terms of a single formula.

Fact. The argument with premises $p_1, p_2, \ldots, p_n$ and conclusion c is valid if and only if the formula $(p_1 \wedge p_2 \wedge \cdots \wedge p_n) \to c$ is a tautology.

As always, take some time to think about this fact and make sure you understand why it is true.

I'd like to talk about one last little bit of notation. It's not strictly necessary to know this in our class,[11] but it's useful to know. Suppose you have an argument with premises $p_1, p_2, \ldots, p_n$ and conclusion c. If that argument is valid, we write $p_1, p_2, \ldots, p_n \vDash c$. The symbol $\vDash$ (LaTeX: `\vDash`, Unicode: `U+22A8`, HTML: `&#8872;`) is called a **double turnstile**, and it comes from a slightly different tradition than the vertical argument notation.

1.12.1 Trivial Validity

Example 1.16. Is the following argument valid?

$$\begin{array}{l} A \wedge B \\ C \\ \hline B \end{array}$$

As you start to build the table, you can see that there are only two assignments that satisfy the first premise $A \wedge B$, and without even looking at the second premise C, we can see that in both of those cases, the conclusion B is also true:

[11] Unless you're in the honors section, in which case you should probably make sure you remember it.

A	B	C	$A \wedge B$	C	B
T	T	T	$\boxed{\text{T}}$		$\boxed{\text{T}}$
T	T	F	$\boxed{\text{T}}$		$\boxed{\text{T}}$
T	F	T	**F**		
T	F	F	**F**		
F	T	T	**F**		
F	T	F	**F**		
F	F	T	**F**		
F	F	F	**F**		

It doesn't even matter what happens with C because any time that first premise $A \wedge B$ is true, the conclusion B must also be true. That's just the nature of $\wedge$. In fact, we're going to take advantage of this kind of rule ("any time $p \wedge q$ is true, so is p, and so is q") in the next section to come up with a new way of proving arguments are valid.

If you really want to see what happens with that second premise we can, but there's nothing that can happen there that would mess up the validity of the argument. In any particular row, if C turns out to be true, then all the premises are true, and we'll need to have a true conclusion, but we already know that the conclusion is true. In any particular row, if C is false, then not all the premises will be true, and we won't care whether the conclusion is true or not.

As it turns out, one of the assignments that satisfies $A \wedge B$ also satisfies C (the first row), and one does not (the second row):

A	B	C	$A \wedge B$	C	B
T	T	T	**T**	**T**	$\boxed{\textbf{T}}$
T	T	F	T	**F**	T
T	F	T	**F**		
T	F	F	**F**		
F	T	T	**F**		
F	T	F	**F**		
F	F	T	**F**		
F	F	F	**F**		

Here's a similar, seemingly simpler question:

Example 1.17. Is the following argument valid?

$$\begin{array}{l} A \wedge B \\ \neg A \\ \hline B \end{array}$$

Let's start to fill out the table:

A	B	$A \wedge B$	$\neg A$	B
T	T	$\boxed{\text{T}}$		$\boxed{\text{T}}$
T	F	**F**		
F	T	**F**		
F	F	**F**		

Here, we have only one assignment that satisfies the first premise, but as in the previous example, that assignment, it also satisfies the conclusion. So by the logic we just used, this must be a valid argument. But if we finish out the table, we see something very strange...

A	B	$A \wedge B$	$\neg A$	B
T	T	T	**F**	T
T	F	**F**	**F**	F
F	T	**F**	T	T
F	F	**F**	T	F

In this case, we have the unusual situation where *none* of the truth assignments satisfy all of the premises. (Or to put it another way, the premises are *inconsistent* with each other.)

Should we still call this argument "valid"? Or should it be "invalid"? Or should there be a third category that is neither valid nor invalid?

"Valid" might seem like an obviously wrong answer because this is a really pointless argument, and it doesn't feel like there's anything "correct" about it. But "invalid" is even worse. Remember that to prove an argument is invalid, we need a counterexample: an assignment that satisfies the premises, but not the conclusion. And we definitely can't find such an assignment here. So there's no "obvious" answer to the question.

Now, before we try to answer this question, I want you to remember that "valid" doesn't mean "good" or "correct". This argument is clearly a stupid argument couldn't ever be useful in the way that more sensible arguments are. We can't apply the argument to any real-world situation, or even to a hypothetical future situation because the situation the premises describe can't ever exist, no matter what meanings you assign to A and B.

This is what we call a **trivial** argument. In mathematics, logic, computer science, and other technical fields, the word "trivial" is used to refer to one of those dumb, annoying edge cases that have to be dealt with, but that don't have much impact on the more normal cases.

For example, in math, exponents are defined for positive integers using a simple rule: b^n is b multiplied by itself n times. But we get a trivial case when $n = 0$. What would it mean to have b multiplied by itself 0 times? The math-users of the world could have decided to declare that $b^0 = 0$, that $b^0 = 1$, or that b^0 is undefined[12] (like dividing by 0). How did they make that decision? Instead of going with what would make the most sense to someone who is just starting to learn how to do exponents, they picked the solution that matches best with everything else people want to do with exponents. We've got patterns like $2^3 = 8$, $2^2 = 4$, $2^2 = 1, \ldots$ where everything decreases by a half. We've got rules like $b^n \cdot b^m = b^{n+m}$, or the formula for the derivative of a polynomial. And we've got dozens of other little properties, patterns, rules, and tricks that all point to $b^0 = 1$ as the most sensible solution. If we'd picked $b^0 = 0$, none of those rules would work any more. Similarly, if we'd gone with undefined, every one of those rules would have to be given a special extra case to deal with the exponent 0.

Fortunately, the people who made this decision centuries ago chose the def-

[12] For reasons that aren't important here, when b and n are *both* 0, 0^0 is still undefined.

inition that is the simplest going forward. And if we work with this definition long enough, it even starts to seem *natural* to treat 1 as the result of multiplying "no numbers" together.

You can see similar things in programming at functions like the string method `.isupper()` in Python, which determines whether every character in the string is in uppercase. If you give it the empty string, or a string of all digits, those are "trivial" cases, and it still has to decide whether to return `True` or `False`. The way that is simplest to code, and the way that preserves the most patterns, rules, and tricks is to return `True`. In other words, if you've got a statement about "all cased characters" and there *are no cased characters*, the simplest answer is to say that the statement is "true". It's *trivially* true, meaning that it's only true in a stupid, technical sense, but it's still easier to accept "true" here than "false".

This kind of claim shows up all over the place: you've got a universal statement that starts out like "every cased character..." or "every assignment that satisfies all the premises..." And you have a trivial case where the "universe" of your universal statement has nothing in it (e.g., there are no cased characters, or there are no assignments that satisfy all the premises). And the way to preserve our rules and patterns is to declare that such a trivial universal statement is always true. If you say "everything" and there are no "things", we're going to call that a true statement. We'll often emphasize that it is only "trivially" true, to drive home the point that we know it's a stupid example, but we still count it as true.

And that's exactly what we're going to do for trivial arguments like the one we were just talking about.

Fact. Any argument whose premises are inconsistent is trivially valid. (In other words, if there are no assignments that satisfy all the premises, then the argument is considered valid. *Trivially* valid, but still valid.)

Eventually, you will get used to this sort of thing, but in the meantime, I think it's helpful to know that this is just a declaration of a technicality. But it's not an *arbitrary* technicality. This choice, and many others like it, make so much of what we do simpler. For example, we have this trick where in order to prove an argument is invalid, you have to find a truth assignment that satisfies the premises and not the conclusion. IF we'd declared trivial arguments invalid, this trick wouldn't work anymore. And the rule (which we'll see in detail in the next section) where we know that under the premise $p \wedge q$, the conclusion p or q will always be true also depends on this technicality.

1.13 Natural Deduction Proofs

Consider the following argument:

$$\begin{array}{l} \neg\neg A \\ A \to (B \wedge C) \\ \hline C \wedge A \end{array}$$

Do you think this argument is valid? Take some time to think about it, but don't build a truth table. Instead ask yourself the following question: If both $\neg\neg A$ and $A \to (B \wedge C)$ are true, does that force $C \wedge A$ to be true as well?

The argument is, in fact, valid. Many students can tell that the argument is valid without actually looking at a truth table or even thinking about truth assignments at all. If I were to ask one of them to explain why this argument is valid, they might say something like this:

> If the second premise $A \to (B \wedge C)$ is true, that means that if A is true, then $B \wedge C$ is also true. But if the first premise $\neg\neg A$ is also true, then that means that A is in fact true, and hence so is $B \wedge C$. That would mean that both C and A were true. So whenever the premises are true, so is $C \wedge A$.

In this section, we're going to take this kind of rough reasoning and make it more precise and formal. That way, we can be sure that every step of the argument is based on sound logical principles. This is what we mean when we say we are writing a "proof": a sequence of logical deductions.

The word **proof** can be used to describe any sequence of reasoning, regardless of how much detail or formality is used. Even abstract, technical fields like theoretical computer science or pure mathematics, there's a lot of variety in how much formality you'll see in a proof. At the most formal end of the spectrum are what we call **formal proofs**, which limit the logical rules to a very specific set of rules that have to be used in a very specific format. Truly formal proofs have almost no explanatory text in them at all. These are the kinds of proofs theoreticians use when they want to prove something *about logic itself*, or when they want to use a computer to help find proofs or to help verify that proofs are correct.

The more ordinary kind of "informal" proof is written in a natural language (such as English) that humans have a much easier time understanding. The types of logical rules that can be used and how much detail needs to be present depends entirely on the intended audience of the proof. Someone writing a proof for an introductory textbook on computational complexity will need to include different details than someone writing up the same proof for publication in *Computer Science Review*, and it would be even more different if they were writing it up for the proceedings of a conference about circuit lower bounds.

In this class, we're not going to write truly *formal* proofs[13] because our ultimate goal is to be able to read and write *informal* proofs of the sort you are likely to encounter in future classes about computing theory. The same general structures appear in both formal and informal proofs, but reading a formal proof is like reading code written without any comments and where all the variables are named `a`, `b`, `c`, etc. You can do it, but it takes a lot of extra time and effort.

By the end of this class, we'll be reading and writing proofs that have a level of detail and formality similar to what will be expected of you in theory classes that you might take in the future. But for the proofs in this section, I'm going

[13] Unless you're in the honors section!

to place some very heavy restrictions on the level of detail you provide and on which logical rules you can use. Even though I are going to be very strict about the requirements, these will not be "formal" proofs.

For starters, we are going to write these proofs using complete English sentences. You'll be allowed some variety in how you choose to phrase the sentences in your proof, but I will be enforcing some rules on what information you need to mention in every step. I call these formal/informal hybrid proofs **semi-formal** proofs. **Semi-formal** is not a term that you'll see outside of this class, but it's useful to have a shorthand term to describe the level of formality that is expected of you in this class.

Why so much detail? A good proof-writer is able to adjust the level of detail in their proofs depending on the needs of their audience. But they must *always* be aware of which details they are leaving out and why. I want you to provide a lot of detail in your proofs right now so that I can be sure you actually know what those details are.

Why limit the set of logical rules? There are a number of extremely important proof strategies that this course is trying to teach. There are often higher-level strategies that can sometimes be used in place of these more basic strategies, but you can't avoid these core strategies all of the time. By restricting the rules you can use, I can make sure you're actually learning these important tools. No matter how many fancy tools a carpenter might have at their disposal, they do still need to know how to use a hammer.

As we move to more general topics, I will gradually relax the constraints on how much detail you need to provide in each proof, but for now, while we're proving things about propositional formulas, I'll expect you to stick to the standards of semi-formal proofs that we'll discuss here.

In the honors section of the course, we'll be doing genuine "formal" proofs, instead of these hybrid "semi-formal" proofs. The structure of the proofs will be virtually identical; the distinction is mostly a matter of presentation. For reasons related to typesetting, I can't include the formal proofs alongside the informal proofs in these lecture notes, so I'll be uploading a separate file to Canvas containing an alternate version of this section, with formal proofs instead of informal proofs.

Let's start by looking at an example of what a semi-formal proof looks like. Let's take that rough explanation from earlier, and make it "semi-formal":

Claim. The following argument is valid:

$$\begin{array}{l} \neg\neg A \\ A \to (B \land C) \\ \hline C \land A \end{array}$$

Proof.

Assume that $\neg\neg A$ and $A \rightarrow (B \wedge C)$ are both true.
Since $\neg\neg A$ holds, A must also hold. (Double Negation)
Because we have A and $A \rightarrow (B \wedge C)$, we also have $B \wedge C$. (Application)
$B \wedge C$ implies C. ($\wedge$-Elimination)
C and A are both true, so $C \wedge A$ is true. ($\wedge$-Introduction)

□

The main strategy being employed here is this: *assume* that your premises are true, use those premises and basic logical rules to infer that other statements/formulas are true, continue to use logical rules to build up more and more true statements until eventually you deduce the desired conclusion. This is the most direct and basic way of proving something. The strategy is so common that it's often just called **proof**, but if you need to distinguish it from other, more indirect methods of proof, it is sometimes called **direct proof**.

This strategy is useful because we are not trying to prove that our conclusion $C \wedge A$ is always true in all circumstances. Typically, we are trying to probe that *under certain assumptions*, the conclusion is true. Every proof you write needs to begin with a line or two where you make those assumptions explicit. Note how the language I used makes it clear that I am not claiming that $\neg\neg A$ and $A \rightarrow (B \wedge C)$ are true. Instead, by using the word "assume", I am setting the stage for the rest of the proof, establishing the *assumptions* that define the imaginary world that the rest of the proof will live inside of.

There are a number of different phrasings that you can use to set up your assumption(s), but you must use some sort of assumption word (such as "assume," "suppose," or "let"). Here are some alternate phrasings for the first line of our proof:

- Assume that $\neg\neg A$ and $A \rightarrow (B \wedge C)$ are both true.
- Suppose $\neg\neg A$ and $A \rightarrow (B \wedge C)$.
- Let $\neg\neg A$ and $A \rightarrow (B \wedge C)$ be true.
- Consider the situation where $\neg\neg A$ and $A \rightarrow (B \wedge C)$ both hold.

After the initial assumption, every step in this proof is essentially the same kind of statement: a deduction that proceeds from previously proven (or assumed) formulas to a new formula that logically follows (always according to some particular rule, but don't worry about the names of the rules yet). I phrased each step of the proof slightly differently to give you an idea of the wide range of possible phrasings for this one simple concept of deduction/inference. There are, of course, many more ways you could phrase a deduction in a proof. For example, all of the following are acceptable ways of phrasing the second line of the proof:

- Since $\neg\neg A$ is true, A must also be true.
- Because we have $\neg\neg A$, we can get A.

- From $\neg\neg A$, we can conclude A.
- $\neg\neg A$ holds, and hence A.
- $\neg\neg A$ leads to A.
- Since $\neg\neg A$, A.

If you want to, you can use exactly the same phrasing on every line. Some proof-writers like to change up the phrasing just for variety's sake. Some people think very carefully about exactly which words will be the most helpful and clearest for each specific situation. And some people have no idea why they prefer certain phrasings in certain situations; they just use what sounds best to them at the time. These details can have subtle effects on how easy it is to read or understand a proof, but they don't affect whether the proof is valid or not. In this class, you can use any phrasing you like, as long as it's clear which previous formulas you are using and what formula you are deducing. We won't deduct points for bad grammar unless it gets so bad that we can't understand what you are trying to say.

Now let's turn our attention those rules. I mentioned earlier that I was going to restrict the set of rules that you are allowed to use (at least when we're being semi-formal about things). Specifically, we are going to restrict ourselves to using the rules from a system called **Natural Deduction**. There are lots of other systems we could take our rules from, but Natural Deduction has two main advantages. First of all, the rules are laid out in a systematic way with exactly two rules for every connective of propositional logic: one for *proving* new formulas that have the connective, and one for *using* previous formulas that have the connective. This makes it easy to remember all the rules, and it simplifies the process of figuring out which proof strategies to use. The second big advantage is that Natural Deduction proofs have pretty much the exact same structure as the informal proofs we'll eventually be writing. So learning to write formal or semi-formal Natural Deduction proofs is basically teaching you how to write "real" proofs as well. There are some common tools that aren't part of Natural Deduction, but the most important ones are all there, so for now (while we're just working with propositional logic formulas), we're going stick to just the rules of Natural Deduction.

All the important rules we need fall into the category of **inference rules**, meaning that they tell you what formulas you can *infer* from the formulas that you already have. For example, the rule called "$\wedge$-Elimination" tells you that if you have a formula of the form $p \wedge q$, then you can conclude p. (You could also conclude q. Or both p and q!)

Keep in mind that the lowercase variables p and q I used to write the rule don't have to be single-letter atomic variables. They could be huge complicated formulas. So if you already have the formula $(A \rightarrow \neg B) \wedge \neg(B \vee C)$, then you could use $\wedge$-Elimination to conclude $A \rightarrow \neg B$. (You could also use it to conclude $\neg(B \vee C)$. Or you could conclude both $A \rightarrow \neg B$ and $\neg(B \vee C)$!) You can use $\wedge$-Elimination on *any* formula that is of the form $p \wedge q$.

But note that while you can fill in p and q with formulas that are as complicated as you like, you can't add extra stuff; the whole formula has to be of the form ____ $\wedge$ ____. Inference rules can't be applied to only *part* of a formula. So for example, if you have the formula $(A \wedge B) \to C$, you can't use $\wedge$-Elimination to conclude $A \to C$ or A or anything like that. $\wedge$-Elimination doesn't apply to the formula $(A \wedge B) \to C$ because it isn't in the form $p \wedge q$. *Part* of it might be in that form, but not the whole thing.

To express these rules succinctly, I'm going to introduce a new symbol: $\vdash$, which is called a "turnstile." It's a relational symbol, like $\equiv$, meaning that it acts like a verb. The symbol $\equiv$ which goes between two formulas to make the claim that those two symbols are logically equivalent. The symbol $\vdash$ (LaTeX: `\vdash`, Unicode: `U+22A2`, HTML: `&#88662;`) goes between a list of formulas (separated by commas) on the left and a single formula on the right, and it makes the claim that you can conclude the formula on the right if you have the formulas on the left. Or to put it another way, "$p_1, p_2, \ldots, p_n \vdash c$" means "there is a proof with assumptions $p_1, p_2, \ldots, p_n$ and conclusion c."

We can use the turnstile to write down claims that we want to prove (e.g., $\neg\neg A, A \to (B \wedge C) \vdash C \wedge B$), but we can also use it to state one specific inference rule (e.g., $p \wedge q \vdash q$). So for example, when I write $p, q \vdash p \wedge q$, I'm saying that if I already have p and I already have q, then I can deduce that $p \wedge q$ is also true.

The Subtle Difference Between $\vdash$ and $\vDash$ You won't be tested on this difference in the regular class, but if you're in the honors section or if you're just interested in this sort of thing, continue reading this paragraph. The symbol $\vdash$ is specifically about proofs, so $p \vdash q$ means "If you assume p, you can prove q." So we use $\vdash$ to talk about what you can *prove*. The symbol $\vDash$ is specifically about *semantics* (for propositional logic, that means *truth assignments*). So $p \vDash q$ means "Every assignment that satisfies p also satisfies q."[14] We use the symbol $\vDash$ to declare that an *argument is valid*. Of course, if we're talking about the any well-established proofs system for propositional logic, having a proof and being a valid argument are the same! If you can write a proof for an argument, then that argument better be valid or there is something deeply wrong with your proof system. In technical terms, this is saying that the proof system is **sound**. A proof system that isn't sound is completely broken and worthless, so this is usually only relevant when you introduce a brand new proof system and you want to show that it actually works. The converse property is more subtle. Ideally, we'd like for every valid argument of propositional to have a proof! Otherwise your proof system isn't **complete**. Not being complete isn't a complete disaster because you can still use the proof system to prove things, but an incomplete proof system means that there are valid arguments that can't be proven using that system. For a well-established proof system like Natural Deduction and a simple logic like propositional logic, soundness and completeness have been established for so long that we rarely ever think about the difference between having a proof and being valid. But if you were making

[14] You can also use the symbol $\vDash$ with a truth assignment and a formula as a shorthand for "satisfies". So $(A = \mathrm{T}, B = \mathrm{F}) \vDash B \to A$ means the same thing as "$(A = \mathrm{T}, B = \mathrm{F})$ satisfies $B \to A$."

up a new proof system, you would need to be able to talk about the difference, and that's why we have two different symbols $\vdash$ and $\vDash$.

The short version: There are subtle technical differences between $\vdash$ and $\vDash$, but if you're just writing Natural Deduction proofs for propositional logic, those differences aren't really important.

Here's a list of the simplest rules from Natural Deduction.

Inference Rule	Name
$p \wedge q \vdash p$ $p \wedge q \vdash q$	$\wedge$-elimination ($\wedge$-E) simplification (simp.)
$p, q \vdash p \wedge q$	$\wedge$-introduction ($\wedge$-I) conjunction (conj.)
$p, p \to q \vdash q$	application (Appl.) modus ponens (M.P.) $\to$-elimination ($\to$-E)
$p \vdash p \vee q$ $p \vdash q \vee p$	weakening (Weak.) addition (add.) $\vee$-introduction ($\vee$-I)
$\neg\neg p \vdash p$	double negation (Dbl. Neg.) $\neg$-elimination ($\neg$-E)

The "introduction" and "elimination" names are the traditional Natural Deduction names. The Natural Deduction names can be confusing in terms of how the rule works, but they are systematic with respect to when you would want to use them. The "introduction" rule for a particular connective is used to prove a new formula with that connective in it. So if you want to prove a formula whose main connective is $\wedge$, you would need to use $\wedge$-Introduction. The "elimination" rule for a connective is useful when you already have proven (or assumed) a formula with that connective. So if you've already got a formula whose main connective is $\wedge$, you would want to use $\wedge$-Elimination.

WARNING: Wanting to use a rule isn't the same as being *able* to use a rule. If your goal is to prove $A \wedge C$, you know that you'll need to use $\wedge$-Introduction, but you can't actually use that rule until you have the formulas A and C by themselves. Similarly, if you have the formula $P \to Q$ as one of your assumptions, you will naturally want to use $\to$-Elimination, but unless you already have the formula P, you can't use that rule!

Of course, this table is not the complete list of Natural Deduction rules. It's missing the "introduction" rules for $\to$ and $\neg$ and the "elimination" rule for $\vee$. Those rules require using subproofs, so we'll talk about them a little bit later. For now, let's focus on these simpler rules.

1.13.1 Conjunction Rules

Some of these rules (such as the $\wedge$ rules) are so simple and obvious that you probably wouldn't even think of them as "rules" at all. But since we're being semi-formal here, pretty much *everything* needs a rule.

Let's start with $\wedge$-Elimination (a.k.a., "Simplification"). This says that if you've got an $\wedge$-formula, then you can derive either one (or both) of the two parts. In our example (in the second-to-last line), we only needed one of the two parts, but I'll show an example later where we need both parts.

In *formal* propositional logic, you can only use $\wedge$-Elimination on one part at a time. If you want to get both parts, you'd have to use two separate steps: one to get the left part, and one to get the right. But in semi-formal proofs, I'll let you combine these into a single step (e.g., "Since $A \wedge \neg B$, we can conclude A and $\neg B$.")

The rule $\wedge$-Introduction (a.k.a., "Conjunction") is also fairly straightforward. If you want to prove that $p \wedge q$ is true, you need to prove that p is true and you need to prove that q is true. We used this in the very last step of the proof we did on page 39.

The names "Simplification" and "Conjunction" are pretty old-school, and you'll usually only see them in older textbooks, especially ones for logic courses taught in philosophy departments.

1.13.2 Application: the Most Important Inference Rule

The third inference rule in the table is actually the most important one. This rule is most often called "Modus Ponens"[15] (especially in philosophy textbooks), but the Natural-Deduction-style name is "$\rightarrow$-Elimination".[16] The computer-sciencey name (and my personal favorite) is "Application"[17].

This rule is all about what we can do when we have an implication formula (either because we proved it in an earlier step, or because we assumed it at the beginning of the proof). By itself, $p \rightarrow q$ doesn't really tell us anything. It says "if p is true, then so is q." Unless we also know that p is true, then $p \rightarrow q$ isn't very useful at all. This is reflected by how the application rule works. We can't use application if all we have is $p \rightarrow q$ by itself. But if, in addition to $p \rightarrow q$, we also know that p is true, then we can get somewhere. In that case, we know that if p is true, then q is true, and we also know that p is true. So that allows us to conclude that q is true. This is exactly how modus ponens / $\rightarrow$-elimination / application works. If you have the implication formula $p \rightarrow q$ and you also have its hypothesis p by itself, then you can get the conclusion q by itself.

1.13.3 Double Negation versus Negation Elimination

Double negation is another pretty clear-cut rule. If you know that $\neg\neg p$ is true (in other words, if you know that p is not false), then you can conclude that p is true. This version of the rule (which allows you to remove a double negation

[15] That's short for "modus ponendo ponens," which is Latin for "method to affirm by affirming." Sometimes, the origin of a term tells you a lot about it. Sometimes it doesn't.

[16] I think that the word "elimination" is very misleading here. It makes you think that this rule will work like $\wedge$-elimination, but that is very much not true, and it's a dangerous way of thinking.

[17] If you think of the atomic propositions as data types (like `integer` or `string`), then you can think of a formula like $p \rightarrow q$ as the signature of a function that takes an argument of type p and returns a value of type q. In this way, the application rule is a lot like function application: if you apply a function with signature $p \rightarrow q$ to an object of type p, you'll end up with an object of type q.

from the front of a formula) is called "$\neg$-elimination" in Natural Deduction. Note that despite the name, it doesn't mean you can eliminate any $\neg$ you see. You can't take $\neg p$ and use that to prove p.[18]

The name "Double Negation" is technically the name of a more powerful version of this rule, which allows you to go in the opposite direction (from p to $\neg\neg p$) or even to apply it to only part of a formula (e.g., from $A \to \neg\neg B$ to $A \to B$). This more powerful rule isn't just a one-direction *inference rule*, but a two-directional *equivalence rule*. But it turns out that once you get all the rest of the Natural Deduction rules, you don't really need the two-directional equivalence version of this rule.

If we were doing truly formal proofs, then I would enforce using the inference rule version, which only allows you to *remove* a double negation, and only when the whole formula is in the form $\neg\neg$____. But for our semi-formal proofs, I'm not going to be that picky, so feel free to apply this rule to parts of formulas or to use it backwards. (Just don't do that for any of the other inference rules!)

1.13.4 Standards for "Semi-Formal" Proofs

This is as good a place as any to talk about the standards I expect for any "semi-formal" proof that you write in this class. Apart from being clear and correct (which is a requirement for all proofs you write anywhere), you should follow the following guidelines:

- Always write down the claim. (It's not part of the proof, but it's important to put it down before you start the proof itself.)
- Clearly indicate the beginning and end of the proof. We often start by writing the word "*Proof:*" or "Pf.", and we often end[19] with a little box off (sometimes called a **tombstone** to the right like this: □
- Clearly indicate all assumptions for the proof.
- Only use one inference rule per step.
- Always cite the rule you use by name. (When there are multiple names for a rule, use whichever one you like best. Feel free to abbreviate.)
- Always cite which formulas you are using in each step. (The words "because" and "since" are useful here.)
- Only use the eight rules from Natural Deduction.

Always remember that proofs are ultimately about communication, and *not* about getting to an answer. So anything you can do to make things clearer will

[18] I hope you're laughing at this. But it's a real mistake that students make sometimes! This is the sort of thing that only happens if you take the names of the rules as descriptions for how they work.

[19] A long time ago proofs used to end with the abbreviation **Q.E.D.**, short for "quod erat demonstrandum", literally meaning "what was to be shown", but nobody does that anymore.

be helpful. For example, since we're citing rules on each step of our semi-formal proofs, it makes things easier if you put each step on its own line.

Now some students always want to know about the bare minimum of writing they can get away with. If you're one of those students, take a look at the following version of the proof we just saw. This is what I would call a minimal example, meaning that if you did any less than this, you'd probably lose points.

Claim. $\neg\neg A, A \to (B \wedge C) \vdash C \wedge A$

Proof.
Assume $\neg\neg A$ and $A \to (B \wedge C)$.
Since $\neg\neg A$, A. (¬-E)
Since A and $A \to (B \wedge C)$, $B \wedge C$. (→-E)
Since $B \wedge C$, C. (∧-E)
Since C and A, $C \wedge A$. (∧-I) □

It's not a very elegant-looking write-up, but it's clear and understandable, and it has all the right bits in the right places.

Before we talk about weakening, let's do another proof that uses these same rules, just for practice:

Claim. $(A \wedge B) \to C, \neg\neg B \wedge A \vdash C$

Proof.
Assume $(A \wedge B) \to C$ and $\neg\neg B \wedge A$.
⋮

I'm going to pause the proof right here. Even though this is an easy problem, I'm going to treat it the same way I would treat a more complicated proof, to demonstrate the kinds of strategies that you can use to find proofs.

For starters, I take note of two things: the formulas that we "have" and the formulas that we "want".

The formulas we "have" are the ones that we've already proven or the ones we've explicitly assumed. Since we just started the proof, all we have are the two assumptions $(A \wedge B) \to C$ and $\neg\neg B \wedge A$. Fortunately, it's easy to keep track of the formulas we have because they are the ones that appear in the proof itself.

The formulas we "want" are those formulas that we would *like* to be able to prove. Right now, this only includes our ultimate goal, the conclusion of the formula: C. But we might add other subgoals to this list in the future. I like to keep track of my goals off in the margins or on a piece of scratch paper. They're not part of the proof itself, but they are important to keep track of, and it can be useful to write them down, especially for more complex problems. So for this problem, I might write down in the margin:

Main Goal: C

Whether you actually write down your goals or not, it is vitally important that you understand the difference between the formulas you have and the formulas you want. Mixing them up will completely destroy your proof, and yet it's one of the most common errors that I see in this course.

Once you have mentally catalogued all the formulas that you have, you can look at the main connective for each of those formulas as inspiration for which elimination rules you are likely going to use. (Similarly, you can look at the main connectives in the formulas that you *want* to figure out which *introduction* rules you'll want to use. At the moment, the only goal formula we have doesn't have any connectives, so there aren't any introduction rules to think about yet).

The first formula we have is $(A \wedge B) \rightarrow C$, and its main connective is $\rightarrow$. So we're going to want to use $\rightarrow$-elimination, also known as "application". Now *wanting* to use a rule doesn't mean we *can* use it. In order to use application, we need both $p \rightarrow q$ (the implication) and p (the premise) by itself. We have the implication, but we don't have the premise $A \wedge B$ by itself, so we can't use application yet. But we *can* add $A \wedge B$ to our list of "wanted" formulas!

Main Goal: C
wanted (for appl.): $A \wedge B$

The second formula that we "have" is $\neg\neg B \wedge A$. Its main connective is $\wedge$, so we want to use $\wedge$-elimination. Fortunately, we don't need any other formulas to use $\wedge$-elimination, so let's do that right now:

Proof.
Assume $(A \wedge B) \rightarrow C$ and $\neg\neg B \wedge A$.
Since $\neg\neg B \wedge A$, both $\neg\neg B$ and A are true. ($\wedge$-E)
⋮

This adds two more formulas to our "have" list: $\neg\neg B$ and A, and it's easy to see that we can use double negation on $\neg\neg B$:

Proof.
Assume $(A \wedge B) \rightarrow C$ and $\neg\neg B \wedge A$.
Since $\neg\neg B \wedge A$, both $\neg\neg B$ and A are true. ($\wedge$-E)
$\neg\neg B$ implies B (Dbl. Neg.)
⋮

This adds B to our "have" list.

At this point, we have at least glanced at all the formulas on our "have" list and done one of three things: used an elimination rule on them ($\neg\neg B \wedge A$ and $\neg\neg B$), added something to our "want" list so that we can maybe use an elimination rule on them later ($(A \wedge B) \rightarrow C$), or we've noted that they are so simple that there's nothing more to "eliminate" (A and B).

But remember that we also added a formula to our "want" list, so we should look at that with "introduction" rules in mind. The main connective in the formula $A \wedge B$ is $\wedge$, so we want to use $\wedge$-introduction to prove it. Of course, this rule requires that we already "have" both A and B. Fortunately, we do!

Proof.

Assume $(A \wedge B) \to C$ and $\neg\neg B \wedge A$.
Since $\neg\neg B \wedge A$, both $\neg\neg B$ and A are true. (∧-E)
$\neg\neg B$ implies B (Dbl. Neg.)
Because A holds and B holds, so does $A \wedge B$. (∧-I)
$\vdots$

And that gives us enough to actually use application and finish the proof.

Proof.
Assume $(A \wedge B) \to C$ and $\neg\neg B \wedge A$.
Since $\neg\neg B \wedge A$, both $\neg\neg B$ and A are true. (∧-E)
$\neg\neg B$ implies B (Dbl. Neg.)
Because A holds and B holds, so does $A \wedge B$. (∧-I)
$(A \wedge B) \to C$, together with $A \wedge B$, gives us C. (Appl.) □

You don't need to go through all that work on every proof. If you see a way through to the end of the proof, take it! But if you don't see it right away, use this strategy to get yourself started. Most of the time, you'll eventually get to a point where the rest of the proof seems clear, and you can just go for it. But until you get to that point, you can and should use this process to guide yourself forward. If you come to ask me questions about how to prove something and you haven't written down a list of your goals, the first thing I'm going to make you do is to write down those goals!

Occasionally, there will be an alternate path to the proof. For example, on this problem if you use the more powerful version of Double Negation (the one that works on parts of formulas), then you could actually use Double Negation *before* you used ∧-Elimination. As long as you know what rule you are using, and as long as I'm allowing you to use that rule, and as long as your proof meets all the requirements for using that rule, then you can use it.

On the other hand, sometimes you'll have a vague feeling that something should be true even though you don't know a rule for it. For example, if you used double negation right at the beginning on this proof and proved $B \wedge A$, you might be thinking "$B \wedge A$ is basically the same as $A \wedge B$, so I should be able to use application now, right?" But that's not how the rule works. In order to use Application, you need to have proven the exact formula $A \wedge B$. If you only have the formula $B \wedge A$, then that's not good enough!

Is there a rule that lets you go from $B \wedge A$ to $A \wedge B$. Well, yes. There is an equivalence rule called "∧-commutativity", but it's *not* part of Natural Deduction, and so it's off-limits for semi-formal proofs. (And even if it were on the list of allowed rules, you would need a separate step to go from $B \wedge A$ to $A \wedge B$ before you could do application.) Your intuition isn't *wrong* here; the correct Natural Deduction proof does involve proving $A \wedge B$ (after an intermediate step or two). Intuitions are great for giving you general ideas of what you should try to prove, and which pieces of information you should use to prove them, but intuitions only provide you with *general ideas*. Each step in a proof (especially a semi-formal proof) needs to be an application of a particular rule, not an intuitive leap.

Of course, even if you knew that this was because of the commutative law for $\wedge$, I've told you not to use that rule, so you might ask why I am being strict about not using that rule, and that's a very reasonable question. I'll try to answer it here:

There are many, many named logical rules out there, and there are infinitely many unnamed, but still logically sound rules. In fact, many of the problems in this class will just be me asking you to prove that some logical rule is a valid one. If I let you use any logically sound rule in your proofs, every proof would end up exactly one step long. So I have to draw the line somewhere. I've decided to (temporarily) draw the line here at the rules of Natural Deduction because those are enough rules to prove every possible valid argument, but there aren't so many of them that we run out of interesting things to prove. If you find these restrictions frustrating, just wait a few weeks and we'll get much more lax about things.

1.13.5 Weakening: The Silly Way to Prove $p \vee q$

Suppose that the formula p is true (presumably you've already proven it, or maybe it was an assumption). If p is true, then it's certainly true that either p or q is true. I mean, it'd be a silly thing to say, but it's certainly true. Why is it silly? Well, p is a much stronger claim than $p \vee q$. If I know that p is true, why would I bother saying that either p or q is true? I know which one it is! Going from the claim that p is true to the claim that $p \vee q$ is true actually *weakens* my claim. This is why the corresponding rule is called "weakening." In Natural Deduction, the rule is called "$\vee$-introduction."[20] The older, philosophical-tradition name for the rule is "addition."[21]

Why would you ever want to use such a rule if it always weakens your claim? Well for starters, I might ask you to prove a weak claim. For example:

Claim. $(X \to Y) \wedge X \vdash Y \vee (W \wedge \neg Z)$

Proof.
Assume $(X \to Y) \wedge X$.
Since $(X \to Y) \wedge X$ holds, $(X \to Y)$ and X are both true. ($\wedge$-E)
We have both $(X \to Y)$ and X, so we can get Y. (Appl.)
From Y, we can derive $Y \vee (W \wedge \neg Z)$. (Weak.) □

In some ways, this use of the weakening rule makes it look very powerful. We went from a simple claim about Y to this huge claim about Y, W, and Z! But if you think about what that huge claim really is saying, you'll quickly realize that while it is a longer formula, it is not a stronger claim. The formula Y just says that Y is true. But $Y \vee (W \wedge \neg Z)$ says that maybe Y is true, or maybe this other weird thing is true.

Does this mean that weakening is only useful for proving stupid statements like this? Not entirely. Many of the more useful applications of weakening

[20] I don't like this name because it makes it sound like $\vee$-introduction works the same way as $\wedge$-introduction, and it doesn't.

[21] I'm not really sure where this name comes from. I mean, I know that $\vee$ is essentially addition in a Boolean algebra, but I don't know why this specific $\vee$ rule (there are others) is the one that gets called "addition."

won't become clear until we start working with subproofs (Proof by Cases in particular), but there are other situations in which we might want a weaker claim than we start with. For example:

Claim. $(A \vee B) \to C, A \vdash C$

In this case, we've got an assumption that is an implication with a weak premise. In order to use Application on the formula $(A \vee B) \to C$, you'll need to explicitly have the formula $A \vee B$. And the only way to get that is using Weakening:

Proof.
Suppose $(A \vee B) \to C$ and A are true.
We can weaken A to get $A \vee B$ (Weak.)
Then we can apply $(A \vee B) \to C$ to $A \vee B$, resulting in C. (Appl.) □

1.13.6 Direct Proof

This general technique of assuming that the premises are true and then proving that the conclusion is also true is called **direct proof**. We've been using it to prove the validity of an entire argument, but the technique is actually more powerful than that. Direct proof can be used to prove the truth of any sort of if-then claim. This includes proving that an argument is valid, but it also includes smaller-scale claims, such as proving that a conditional formula (i.e., a formula using →) is true.

For example, suppose we wanted to prove that $(A \wedge B) \to C, A \vdash (B \wedge D) \to C$. We could start our proof as normal:

Claim. $(A \wedge B) \to C, A \vdash (B \wedge D) \to C$

Proof.
Assume $(A \wedge B) \to C$ and A.
$\vdots$

But after that first assumption, we immediately run into some problems. The first strategy you should follow is to think about the formulas that we have to work with (i.e., the formulas we've assumed or already proven. In this case, we have that single formula A, which isn't really useful on its own, and we have the conditional $(A \wedge B) \to C$, which is only useful if we can somehow get $A \wedge B$. (If we had $A \wedge B$, then we'd be able to use the application rule.) But $A \wedge B$ isn't going to happen anytime soon because even though we know that A is true, we don't know anything about whether B is true.

The second important strategy is to think about the formulas that we want to prove. Our goal is to prove a different conditional: $(B \wedge D) \to C$. We've got a rule (∧-introduction) for proving conjunction formulas, and we've got one for proving disjunction formulas (weakening), but we don't have any sort of →-Introduction rule yet.

So what would an →-Introduction rule even look like? Let's think about what it really means to know that an implication formula is true. One way to interpret the formula $(B \wedge D) \rightarrow C$ is: "*if* $B \wedge D$ is true, *then* C is true." Another way to phrase this would be to say that $B \wedge D$ *implies* C." Note that this is *not* the same as saying that $B \wedge D$ is true. $(B \wedge D) \rightarrow C$ *sort of* makes the claim that C is true, but it only makes that claim *under the condition* $B \wedge D$. In other words, $(B \wedge D) \rightarrow C$ doesn't mean anything by itself. But *if* someone else could establish that $B \wedge D$ was true, *then* $(B \wedge D) \rightarrow C$ would make the claim that C was true. If $B \wedge D$ is not true, then $(B \wedge D) \rightarrow C$ wouldn't be making any claim at all.

So establishing that $(B \wedge D) \rightarrow C$ is true does not require us to establish that $B \wedge D$ is true at all! It does require that we establish that C is true, but even then, we only need to establish that C is true *in the situation where we already know* $B \wedge D$. What we really need to do is to temporarily add $B \wedge D$ to our assumptions, figure out how to use that to prove C is true, and then we can conclude that $B \wedge D$ *implies* C.

This should sound familiar. The process of assuming some kind of premise and then proceeding to prove that some conclusion is true is what we've done in every proof problem so far. This is exactly what **direct proof** is intended to do. To prove $(B \wedge D) \rightarrow C$, we need write a whole new (smaller) proof (a **subproof**) where we add $B \wedge D$ to our assumptions and then proceed to prove C *inside the subproof.* Then when we leave the subproof and return to the main proof, we know that the new assumption $B \wedge D$ *leads to* the conclusion C. In other words, we know that $B \wedge D$ *implies* C.

In a semi-formal proof, you would write it like this:

Claim. $(A \wedge B) \rightarrow C, A \vdash (B \wedge D) \rightarrow C$

Proof.

Assume $(A \wedge B) \rightarrow C$ and A.

    Suppose that $B \wedge D$ is true.

    Since $B \wedge D$, we have B. (∧-E)

    Since B and A are both true, $A \wedge B$ is also true. (∧-I)

    Because we have $(A \wedge B) \rightarrow C$ and $A \wedge B$, we get C (Appl.)

Assuming $B \wedge D$, we proved C, so $(B \wedge D) \rightarrow C$. (dir. pf) □

Here's the rule expressed more formally:

Inference Rule (Direct Proof)**.** If you have a subproof with assumption p and conclusion q (inside the subproof), then you can conclude $p \rightarrow q$ is true outside of the subproof.

In Natural Deduction, "Direct Proof" is called **→-Introduction**, because this is the rule that you would use if you wanted to prove a formula where → is the main connective.

One thing that causes students to feel a little lost the first time they do a direct proof as a subproof is that the assumption of the subproof ($B \wedge D$ in this case) seems to come out of nowhere. And it really does! Just like the initial assumptions of the main proof, we basically just chose the assumption that matches the premise of our goal. You're allowed to assume any formula you want when starting a subproof. Unfortunately, this option to assume whatever you want is frequently abused by students. It's *legal* to start a subproof with any assumption you can imagine, but your choice of assumption is going to change the formula that you end up with when you leave the subproof and head back to the main proof.

Here's a useful rule of thumb: When choosing your assumptions, always THINK ABOUT WHAT YOU ARE GOING TO DO WITH THE SUBPROOF WHEN YOU'VE FINISHED IT. In the above example, our goal was to prove $(B \wedge D) \rightarrow C$, which is only possible if we assume $B \wedge D$ and prove C.

A Common Mistake with Direct Proof

New proof-writers often make the mistake of thinking about what they are going to do *inside* the subproof instead of thinking about what they're going to do *after* the subproof is over. If you made this mistake in the above example, you might have been thinking something along the lines of this:

> "In order to use $(A \wedge B) \rightarrow C$, I'm going to need to have both A and B. I already have A, so it'd be nice if I had B."

So far this is a very reasonable chain of thought. But here's where some students start to drift off course.

> "Since it'd be nice if I had B, why don't I go ahead and assume B in a subproof?"

And so they might start their proof like this:

Proof.

Assume $(A \wedge B) \rightarrow C$ and A.

    Suppose that B is true.

    ⋮

Now this proof doesn't actually contain any invalid steps yet. But it's already drifting off course, heading towards a completely different destination than the one we were hoping for. We could continue in this general direction without introducing any more errors, but without getting closer to our true goal:

Proof.

Assume $(A \wedge B) \rightarrow C$ and A.

    Suppose that B is true.

    Since B and A are both true, $A \wedge B$ is also true. ($\wedge$-I)

    Because we have $(A \wedge B) \rightarrow C$ and $A \wedge B$, we get C (Appl.)

    ⋮

Now at this point, it's time to end the subproof and use the Direct Proof rule to summarize into a single formula what this subproof has shown us. There are two things that can happen here. If you understand how the direct proof rule works, you will end up with this completely correct proof:

Proof.

Assume $(A \wedge B) \to C$ and A.

Suppose that B is true.

Since B and A are both true, $A \wedge B$ is also true. ($\wedge$-I)

Because we have $(A \wedge B) \to C$ and $A \wedge B$, we get C (Appl.)

Assuming B, we proved C, so we can conclude $B \to C$. (dir. pf)

Uh-oh... This is not what I was trying to prove...

This is not such a horrible place to be, as it's now obvious that the assumption we made didn't help us to prove $(B \wedge D) \to C$, which is the goal we actually wanted to prove. We can back up to the subproof assumption, assume $B \wedge D$, and finish the proof correctly.

Unfortunately, what happens more often is that the proof writer either just assumes that they proved their goal, even though they actually proved something else:

Proof.

Assume $(A \wedge B) \to C$ and A.

Suppose that B is true.

Since B and A are both true, $A \wedge B$ is also true. ($\wedge$-I)

Because we have $(A \wedge B) \to C$ and $A \wedge B$, we get C (Appl.)

Make up some mumbo-jumbo and conclude $(B \wedge D) \to C$. ERROR!

The only logical error here is literally in the last step. But that logical error is really only because they started on the wrong track with their assumption and then tried to "fix" it without going back and picking the correct assumption.

Sometimes a proof-writer will get to this point, and they'll sense that something is wrong without really being able to put their finger on what the problem is. So they'll do try to patch the problem by adding extra stuff to the proof, often by trying to add more lines to the subproof, possibly trying to prove both $B \wedge D$ *and* C. But that doesn't fix the real problem, which is that they made a not-helpful assumption in the first place.

What is the moral of this story? Well, you can try to avoid the wrong turn in the first place by choosing your assumption based on what you are going to do *after* you're finished with the subproof and not based on what you are going to do *inside* the subproof. But there's a secondary lesson here too, which can help you out if you accidentally choose the wrong assumption. After you have ended the subproof and you are using the subproof to prove a new formula LOOK AT WHAT YOU ACTUALLY ASSUMED AND WHAT YOU ACTUALLY PROVED and use that information to figure out the formula that you have actually proved. Then you can compare this to what you wanted to prove, and if they don't match up, you can go back and try a different assumption.

Phrasing for Direct Proofs

This is why I am forcing you to always summarize the proof by explicitly stating your assumption and your conclusion, when you are using the Direct Proof rule. Now I'm not going to enforce any one particular way of phrasing this, but you must always summarize your subproof in a way that makes it clear what the assumption of the subproof was (what did you assume to start the subproof?) and what the conclusion of the subproof was (what was the last thing you proved in the subproof?). Here are several different ways you could phrase this information:

- Assuming $B \wedge D$, we proved C, so $(B \wedge D) \to C$.
- Given $B \wedge D$, we showed C, so $(B \wedge D) \to C$.
- From $B \wedge D$, you can get C, thus $(B \wedge D) \to C$ holds.
- Because $B \wedge D$ implies C, $(B \wedge D) \to C$ is true.
- If $B \wedge D$ holds, then so does C, and hence $(B \wedge D) \to C$.
- I proved C under the assumption $B \wedge D$. Therefore $(B \wedge D) \to C$.
- Since C is true given $B \wedge D$, we have $(B \wedge D) \to C$.
- In the case where $B \wedge D$ holds, we get C, and so we have $(B \wedge D) \to C$.

Note that it would *not* be correct to write something like "Since we proved $B \wedge D$ and C, we can conclude $(B \wedge D) \to C$." While we did prove C (at least we proved it under the assumption $B \wedge D$), we absolutely did not "prove" $B \wedge D$. We *assumed* $B \wedge D$. So make sure that the words you use make it clear that you *assumed* $B \wedge D$ and that you proved C *under that assumption.*

Which formulas are accessible in a subproof?

Inside a subproof, it's always acceptable to use formulas that were assumed or proven earlier in the main proof. This is what makes it a *sub*proof, and not just a separate proof. You'll notice that I did that in the last two lines of the subproof. If we were using a more formal system for writing proofs, you'd probably be required to use a "reiteration" rule to bring those formulas "into" the subproof on their own lines before using them in the subproof. But in this class, as long as you're using the formulas correctly, you can always use formulas from the main proof in any subproof.

WARNING: You may *not* take formulas *out* of a subproof. Once a subproof is over, you can't use those formulas in the main proof.

Let's do another example, just for practice.

Claim. $(X \vee Y) \to \neg\neg Z \vdash X \to Z$

Proof.

Suppose $(X \vee Y) \to \neg\neg Z$.

  Assume X.

  Since X is true, so is $X \vee Y$. (Weak.)

  Because we have $X \vee Y$ and $(X \vee Y) \to \neg\neg Z$, we can conclude $\neg\neg Z$. (Appl.)

  From $\neg\neg Z$, we can conclude Z. (Dbl. Neg.)

We've shown Z is true given X, and therefore $X \to Z$. (dir. pf)

One more example, not because it's difficult, but because it often makes people panic when they see it.

Claim. $\vdash (\neg\neg P \wedge Q) \to (Q \wedge P)$

What does this even mean? There's nothing to the left of the turnstile, and that makes some students panic at first sight. But it means more or less what you might expect. If there are no formulas to the left of the turnstile, that just means that you can prove this formula is true without having any assumptions in the main proof.

We write the proof just as we always do, only we don't write down any assumptions for the main proof. Of course, since our goal is an → formula, we'll need a subproof, and the *subproof* will have an assumption, but that's something we can handle in a moment. We'll just start by jumping directly into the subproof, like so:

Proof.

Assume $\neg\neg P \wedge Q$.

Since $\neg\neg P \wedge Q$, we have both $\neg\neg P$ and Q. (∧-E)

$\neg\neg P$ implies P. (Dbl. Neg.)

From P and Q, we get $Q \wedge P$. (∧-I)

From $\neg\neg P \wedge Q$, we have $Q \wedge P$. Hence $(\neg\neg P \wedge Q) \to (Q \wedge P)$. (dir. pf) □

1.13.7 Proof by Contradiction

Now that we have the hang of subproofs, let's see if we can use them for something other than proving implications. What if we wanted to prove a negation? How can we prove a formula like $\neg p$? (Remember, p is *not* necessarily an atomic formula. It could be (and usually is) a much bigger formula.) The tactic we're going to use here is similar to one that you might use in any ordinary debate. To prove that some particular claim is wrong, you pretend (for the sake of argument) that it *is* true and you show how this fact leads to something else untrue (usually a contradiction).

The "pretend that it's true" part works exactly like the subproofs we've already seen. If we want to prove $\neg p$, we assume p. This may seem weird, especially considering we actually think p is going to be false, but that's sort of the whole point of this kind of proof. *We* think that p is wrong, but in order to convince someone else, we show how believing p leads to believing something impossible (something that *everyone* can agree is impossible). Now everyone agrees that it's impossible to have a formula q and its negation $\neg q$ both be true, so a formula like $q \wedge \neg q$ would be a perfect target for proving "something impossible". Notice that we don't really care what q is, as long as we can prove (in the hypothetical universe where p is true) both q and $\neg q$.

Let's take a look at a specific example:

Claim. $A \to B \vdash \neg(A \wedge \neg B)$

Proof.

Let $A \to B$ be true.

⋮

We set up our assumption and then set as a goal the formula $\neg(A \wedge \neg B)$. The assumption $A \to B$ is an → formula, so we're probably going to want to do Application at some point, but we can't do that right now because we don't know that A is true. We can add A to our list of goals because it would be nice to have A, but that's about it.

Since our main goal is to prove that $A \wedge \neg B$ is *false*, we are going to use Proof by Contradiction. So we start by assuming that it's *true*, and we are going to use that to prove something contradictory. We don't actually believe that $A \wedge \neg B$ is true; we're just assuming it temporarily to prove how it must lead to a contradiction (and hence must be impossible). You should mentally be reading the contradiction subproof in a sarcastic voice.

Proof.

Let $A \to B$ be true.

  Assume that $A \wedge \neg B$ is true (will show a contradiction).

  ⋮

When you start a contradiction proof, you should make it clear that you are doing a contradiction proof and not a direct proof. Here are some alternate ways to phrase the first line of this contradictions subproof.

- Suppose towards a contradiction that $A \wedge \neg B$ were true.
- Assume $A \wedge \neg B$. I will show a contradiction.
- Let $A \wedge \neg B$ be true. We will prove a contradiction.
- For the purposes of contradiction, suppose $A \wedge \neg B$.

The phrase "suppose towards a contradiction" is an odd little phrase, but it's very common in proofs. Just make sure you are using it correctly. If you don't exactly get the wording correct, you'll either end up with nonsense or something that is just wrong (e.g., "Suppose a contradiction…").

Alright, so we've set an assumption for the subproof. Now it's time to set the subgoal of the subproof. Unlike with Direct Proof, we don't have a *specific* formula in mind for our goal. What we want is just any sort of contradiction. We want to prove that some formula is true and that it's negation is true. Maybe we'll end up proving A and $\neg A$ are both true. Maybe it will be B and $\neg B$. It's unlikely, but it might even be something like $A \vee B$ and $\neg(A \vee B)$.

Getting back to our proof, we now have an $\wedge$ formula, so we can do $\wedge$-Elimination. And that's nice because it gives us the formula A that we were waiting on earlier. So we can do Application too:

Proof.

Let $A \to B$ be true.

  Assume that $A \wedge \neg B$ is true (will show a contradiction).
  $A \wedge \neg B$ implies A and $\neg B$. ($\wedge$-Elim.)
  Since $A \to B$ and A, B. (Appl.)

  ⋮

And there's the contradiction we're looking for! We can't have both B and $\neg B$, so our last assumption $A \wedge \neg B$ must be *false*.

Proof.

Let $A \to B$ be true.

Assume that $A \wedge \neg B$ is true (will show a contradiction).

$A \wedge \neg B$ implies A and $\neg B$. (∧-E)

Since $A \to B$ and A, B. (Appl.)

Assuming $A \wedge \neg B$, we proved B and $\neg B$, which contradict each other, and therefore $\neg(A \wedge \neg B)$. (contrad.) □

Just like with Direct Proof subproofs, you should always[22] summarize the subproof, making sure to mention the assumption and the conclusion (the two contradictory formulas). Here are several different ways you might phrase this.

- But P and $\neg P$ can't both be true, so the assumption $Q \to P$ must be false; hence $\neg(Q \to P)$.
- The assumption X led to Y and $\neg Y$, which is impossible. Therefore $\neg X$.
- Assuming $A \wedge \neg B$, we proved B, which contradicts $\neg B$, so we can conclude $\neg(A \wedge \neg B)$.
- Under the assumption $A \vee B$, we proved C, which contradicts our earlier assumption $\neg C$. Therefore $\neg(A \vee B)$.
- By assuming X, we were able to prove $X \vee Y$, which is inconsistent with $\neg(X \vee Y)$, which we proved earlier. Therefore $\neg X$.

Of course, it should go without saying that you should only use phrases like "earlier assumption" or "we proved earlier" when that is correct for the formula in question. In the example we just did, neither of these would be appropriate because we proved both B and $\neg B$ at the end of the subproof. Neither was an earlier assumption, nor were they statements proven before the subproof.

Here's one way you could phrase the rule Proof by Contradiction:

Inference Rule (Proof by Contradiction)**.** If you have a subproof with assumption p and two contradictory conclusions q and $\neg q$ (i.e., if $p \vdash q \wedge \neg q$), then $\neg p$ is true (i.e., $\vdash \neg p$).

In Natural Deduction, this rule is called **¬-Introduction**. And the old-school philosophy-style Latin name for the rule is **reductio ad absurdum**[23].

A lot of the formulas we've proven so far have been silly little things that are mostly useful as ways to demonstrate these various rules. So let's use these tools to prove something actually useful. You may remember that every implication is equivalent to its **contrapositive**. In other words $p \to q \equiv \neg q \to \neg p$. You

[22] As long as we are being semi-formal. In informal proofs, you only need to summarize when it makes the proof easier to read, such as when the contradictory statements are not immediately obvious.

[23] Which literally means "reduction to the absurd" in Latin. These days "absurd" often means "silly" (think *Monty Python*), but the older meaning of the word is more like "obviously wrong". The rule name is often just shortened to "reductio."

proved this in a homework assignment using tables, but we can also prove that this is true (at least in one specific example) using a Natural Deduction proof. (Note that to fully prove the equivalence, you'd also have to write a proof of the claim $\neg B \to \neg A \vdash A \to B$, but this is a good start.)

Claim. $A \to B \vdash \neg B \to \neg A$

Proof.
Let $A \to B$ be true.
$\vdots$

At this point our main goal is to prove the formula $\neg B \to \neg A$. All I have to work with is the formula $A \to B$. So you might be thinking that A would be a nice formula to have, because you could then use application with $A \to B$. But that is *not* a good reason to assume A. Never start a subproof unless you know exactly what rule you're going to use when you finish the subproof. In this case, if you assume A for a direct proof, you'd end up proving $A \to$ something, which is not what we want. We can't actually do anything with $A \to B$ right now other than to add A to our list of goals as a sort of "it'd be nice to have" entry.

We are going to start a Direct Proof subproof here, but it has nothing to do with our assumption $A \to B$. Instead, it's because our *goal* is an implication. We want to prove $\neg B \to$ something, so Direct Proof is the way to go, and for that, we should assume $\neg B$.

Proof.
Let $A \to B$ be true.
Assume $\neg B$.
$\vdots$

Since we're going to use direct proof at the end of this subproof to prove $\neg B \to \neg A$, our subgoal for this subproof is to prove $\neg A$. Note that this subgoal is a *negated* formula, so that means we should try a Proof by Contradiction. I'm putting the whole direct proof on the back-burner, meaning that I'm not going to think about it at all until I'm done with this contradiction stuff.

So if my goal is to prove (via contradiction) $\neg A$, i.e., that A is *false*, then really my goal is to show that if you believe A, then you'll believe anything, including something impossible. So we start by assuming A.

Proof.
Let $A \to B$ be true.
Assume $\neg B$.
Suppose A (towards a contradiction).
$\vdots$

And my subsubgoal for this subsubproof is some sort of contradiction. I need to prove some formula and its negation. Which formula is up to me. What do I have to work with? Well, I've got A and I've got $\neg B$, which aren't much use on their own (unless I can get $\neg A$ or B). I've also still got $A \to B$, which is also useless, unless I also have A, which– hey, I *do* have A!

Proof.

Let $A \to B$ be true.
  Assume $\neg B$.
    Suppose A (towards a contradiction).
    Since $A \to B$ and A, we get B. (Appl.)

$\vdots$

Great! So now I've met my subsubgoal of finding a contradiction. So now it's time to back out of the subsubproof and use the Proof-by-Contradiction rule. My assumption was A, and I have proven a contradiction, so that means A can't actually be true.

Proof.
Let $A \to B$ be true.
  Assume $\neg B$.
    Suppose A (towards a contradiction).
    Since $A \to B$ and A, we get B. (Appl.)
  Under the assumption A, we proved B, which is impossible because we previously assumed $\neg B$. Therefore $\neg A$. (contrad.)

$\vdots$

Now where was I...? Right, I was in the middle of a subproof (for a Direct Proof), trying to prove that $\neg A$ is true. That's convenient, because I just *did* prove $\neg A$! So now I can back out of the subproof and use the Direct-Proof rule. My assumption was $\neg B$, and I finished by proving $\neg A$, so that means I've proven $\neg B \to \neg A$. Again, that's very convenient because that's exactly what I was trying to prove!

Proof.
Let $A \to B$ be true.
  Assume $\neg B$.
    Suppose A (towards a contradiction).
    Since $A \to B$ and A, we get B. (Appl.)
  Under the assumption A, we proved B, which is impossible because we previously assumed $\neg B$. Therefore $\neg A$. (contrad.)
We've shown $\neg A$ given $\neg B$, and so we've shown $\neg B \to \neg A$. (dir. pf) □

1.13.8 Proof by Cases

If I get a chance, I'll provide more detail later, but I wanted to get the minimum details uploaded as soon as possible.

Let's look at a motivating example, and see how far we can get until we run into a task we don't have the tools for.

Claim. $(P \to Q) \vee \neg\neg Q, P \wedge R \vdash Q \wedge R$

Proof.
Assume $(P \to Q) \vee \neg\neg Q$ and $P \wedge R$.
From $P \wedge R$, we can conclude P and R. ($\wedge$-Elim.)

$\vdots$

We can get our assumptions down, and we can set our main goal $Q \wedge R$. Now we don't have any sort of $\vee$-Elimination rule, so there's not much we can do with the assumption $(P \to Q) \vee \neg\neg Q$ yet. Let's put that on hold and see if there's anything else we can work with.

Our other assumption was $P \wedge R$, which has main connective $\wedge$, so we can do $\wedge$-Elimination to get P and R. But that's as far as we can get along those lines.

So the next order of business is to look at our goal(s). We currently have one main goal: the $\wedge$-formula $Q \wedge R$. In order to prove that, we'll need to use $\wedge$-Introduction, and in order to do that, we'll need to somehow find a way to prove both Q and R. We've already got R, so that's nice, but we still need Q. So we can add Q to our intermediate goals list.

Presumably, we'll have to find some way to use the $\vee$-formula $(P \to Q) \vee \neg\neg Q$ to prove Q. So let's think about what it would look like to have a rule that lets you *use* an "or" statement. Unfortunately, "or" statements are very weak statements. We don't know whether $P \to Q$ is true or not. We don't know whether $\neg\neg Q$ is true or not either. Or maybe it's both cases! Who knows? Not us! All we really know is that *at least one* of those two cases must be correct. Is that enough for us to determine that Q is true?

If you think about it, it does seem like that should be enough. I mean if the first case is correct (i.e., if $P \to Q$ *were* true), then we could combine that with P to get Q (using Application). And if the second case is correct (i.e., if $\neg\neg Q$ were true), then Q would be true because of Double Negation. We know at least one of those cases must be correct, so at least one of those arguments must be applicable. We basically have two reasons why Q is true, and while we have no idea which reason is the correct reason, we do know that at least one of them must be correct. So regardless of which case we're in, we know that Q is true.

This kind of dual reasoning is the basis for the rule we call **Proof by Cases**. Here's one way you could write that down as a semi-formal proof:

Proof.

Assume $(P \to Q) \vee \neg\neg Q$ and $P \wedge R$.
From $P \wedge R$, we can conclude P and R. ($\wedge$-Elim.)
    Case 1: Assume $P \to Q$
    Since $P \to Q$ and P, Q. (Appl.)
    Case 2: Assume $\neg\neg Q$.
    From $\neg\neg Q$, we get Q. (Dbl. Neg.)
In either case of $(P \to Q) \vee \neg\neg Q$, we proved Q, and so Q must be true in general. (Cases)
Q and R are both true; hence, so is $Q \wedge R$ ($\wedge$-Intro.) □

The phrase "in general" is being used here to indicate that the formula Q is true outside of the cases/subproofs. Normally, a formula proven inside of a subproof is *not* true "in general"; it's only true *inside* the subproof. If all we had were these two cases/subproofs proving Q, then we would *not* be able to prove that Q was true in general (outside of the subproofs). But since we have the two cases/subproofs *and* we have the formula $(P \to Q) \vee \neg\neg Q$ (telling us that one of these cases must be correct), we can then conclude Q is also true *outside* of the subproofs.

In this proof, we showed that Q was true in both proofs, concluded that Q was true in general, and then used that to conclude $Q \wedge R$. We also have the option of proving the entire main goal $Q \wedge R$ in both cases, which makes for a very slightly longer proof, but is otherwise just fine.

Proof.

Assume $(P \to Q) \vee \neg\neg Q$ and $P \wedge R$.	
From $P \wedge R$, we can conclude P and R.	($\wedge$-Elim.)
Case 1: Assume $P \to Q$	
Since $P \to Q$ and P, Q.	(Appl.)
Q and R are both true; hence, so is $Q \wedge R$	($\wedge$-Intro.)
Case 2: Assume $\neg\neg Q$.	
From $\neg\neg Q$, we get Q.	(Dbl. Neg.)
Q and R are both true; hence, so is $Q \wedge R$	($\wedge$-Intro.)
Under both assumptions ($P \to Q$ and $\neg\neg Q$), we proved $Q \wedge R$. Therefore $Q \wedge R$ is true.	(Cases) □

Here is one way to write the rule.

Inference Rule (Proof by Cases)**.** If you have a formula of the form $p \vee q$ (i.e., $\vdash p \vee q$) and if you have subproofs with assumptions p and q, each with the same conclusion r (i.e., $p \vdash r$ and $q \vdash r$) , then this is enough to prove that r is true in general (i.e., $\vdash r$).

The Natural Deduction name for Proof by Cases is **$\vee$-Elimination**. But remember not to take the word "elimination" too seriously here. This is a rule about how to *use* "or" formulas, not a rule about how to "eliminate" the $\vee$ symbol.

As far as formatting and phrasing is concerned, you should always label the two assumptions as "cases" (or "options" or other word with the same meaning). This tells the reader why you are making these assumptions and it also helps separate the two subproofs, which is very important because the indentation won't help you keep them separate. For semi-formal proofs, make sure you explicitly reference the $\vee$ formula that makes the Proof by Cases possible (or the two parts of the $\vee$ formula). You can phrase this in several different ways. Here are some alternate phrasings of the last line in the above proof:

- In either case ($P \to Q$ or $\neg\neg Q$), $Q \wedge R$ is true.
- Since and $Q \wedge R$ is true in both cases of $(P \to Q) \vee \neg\neg Q$, it is true in general.
- If either $P \to Q$ or $\neg\neg Q$ is true, then $Q \wedge R$ is true, so $Q \wedge R$ must be true.
- In both cases ($P \to Q$ and $\neg\neg Q$), we proved $Q \wedge R$, so it must be true.

Note that this is the only rule for which you should start a subproof based on formulas that you *have* instead of based on formulas that you *want.* But my earlier advice about what to think about when choosing your assumptions still holds. Always think about what you are going to do *after* the subproof(s) are over. If your goal is a formula like $p \to$(something), then a Direct-Proof subproof is a good idea, and you should make sure you are assuming p, so that when you're done with the subproof, you end up actually proving $p \to$ (something). If your goal is a formula like $\neg p$, then when you start your Contradiction subproof, you should assume p is true. That way, when you've found your contradiction the subproof is over, you end up proving $\neg p$.

If your goal is $p \vee q$, then that does *not* mean you should start a Proof by Cases! We don't use Proof by Cases to *prove* that an $\vee$ formula is true. (That's what Weakening is for.) You should only start up Proof by Cases if you *already know an* $\vee$ *formula.* Because when you finish those cases, you will only be able to use them if you combine them with an $\vee$ formula that you already know.

Sometimes you will coincidentally end up proving an $\vee$ formula using Proof by Cases, just like in the last example, we proved an $\wedge$ formula using Proof by Cases. You can get any kind of formula out of Proof by Cases, but only if you *already have* an $\vee$ formula as one of your assumptions or as something you proved earlier. Here's an example:

Claim. $P \to Q, P \vee (R \wedge S) \vdash Q \vee R$

Proof.
Assume $P \to Q$ and $P \vee (R \wedge S)$.
One possibility is that P is true.
In that case, we can apply $P \to Q$ to P to get Q. (Appl.)
From Q we can get $Q \vee R$. (Weak.)
The other possibility is that $R \wedge S$ is true.
In that situation, we can use $R \wedge S$ to deduce R. ($\wedge$-Elim.)
R gives us $Q \vee R$. (Weak.)
Under either assumption (P or $R \wedge S$), we proved $Q \vee R$, so that must be true in general. (Cases)

Here's an example where you need both a direct proof and a Proof by Cases:

Claim. $Y \to X \vdash (X \vee Y) \to X$

Proof.
Assume $Y \to X$.
Assume $X \vee Y$.
Case 1: Assume X.
We are done.
Case 2: Assume Y.
$Y \to X$ and Y imply X. (Appl.)
In either case of $X \vee Y$, X is true. (Cases)
From $X \vee Y$, we proved X; and therefore, $(X \vee Y) \to X$ is true. (dir. pf) □

Note the phrasing on Case 1. The assumption of the subproof needed to be exactly the same as its conclusion, so there was no need to use any logical inference rules there, but I needed to say something. It's common in real-world proofs to use a phrase like "We are done," or "There is nothing to show," for cases that are completely trivial like this.[24]

1.13.9 Guidelines and Strategies for Semi-Formal Natural Deduction Proofs

Let's summarize the rules for how much detail I expect from your semi-formal subproofs.

Guidelines for Subproofs in Semi-Formal Proofs

- Always indent subproofs.
- When using the direct proof rule, always refer to the entire subproof, including its assumption and conclusion (e.g., "Assuming P, we proved Q, therefore $P \to Q$.")
- When starting a Proof by Contradiction, indicate that you are starting a contradiction subproof (e.g., "Assume towards a contradiction that P is true.")
- When using a Proof by Contradiction, mention the assumption and point out the two contradictory formulas (e.g. "But P and $\neg P$ (which we proved under the assumption Q) contradict each other.")
- Clearly indicate that the two subproofs of a Proof by Cases are being used for Proof by Cases (e.g., "Case 1: Assume P.")
- When finishing a Proof by Cases, always refer to the two parts of the $\vee$ formula and the formula that you proved in both cases (e.g., "In either case (P or Q), R is true.")

The Constructive Strategy

You may have noticed that I've been following a particular strategy when it comes to these Natural Deduction proofs. Here is that strategy laid out a bit more explicitly:

1. Write down all of your assumptions (if there are any).
2. Write down your main goal. Usually, there's no need to mention your list of goals in the proof itself, but it's helpful to keep track of them separately.
3. Look at the formulas you *have* (i.e., the ones you have assumed and the ones you have already proven) and consider their main connectives. (Ignore any secondary connectives and just focus on the main connective.) The formulas you already know tell you which "Elimination" rules you are going to want to use.

[24] In real-world *informal* proofs, the whole case is often reduced to one sentence, like "In the first case, there is nothing to show."

4. Look at the formulas in you *want* (i.e., the ones in your goal list), and consider their main connectives. (Again, ignore any secondary connectives.) The formulas you want to prove tell you which "Introduction" rules you are going to want to use.

    - If you *can* use one of these rules, do so.
    - If you need other formulas to use one of these rules, put this rule on hold and add those other formulas to your goal list. (I usually label them "Want:", and if I think I might forget why I wanted them, I might add a comment to remind me why I wanted them.) Remember that there is nothing you can do right away that will automatically give you these formulas. In particular, you should not start a subproof by assuming a formula that you need. Just make a note, and hopefully the formula you want will eventually show up as a consequence of something else you do. THIS IS ESPECIALLY IMPORTANT FOR THE APPLICATION RULE.
    - If one of these rules requires a subproof (e.g., Direct Proof, Proof by Contradiction, or Proof by Cases), start the subproof with the appropriate assumption, and add the appropriate subgoal to your goal list. (I usually label these "Subgoal:".

5. When you've added a new formula to the goal list, go back to step 3 to see if there's a new "Introduction" rule you can try.

6. When you've proven or assumed a new formula, go back to step 2 to see if there's a new "Elimination" rule you can try. You should also check to see if the new formula fills one of your goals.

    - If the new formula fulfills one of your "Wants", then you may be able to carry out one of those rules you put on hold earlier.
    - If the new formula fulfills one of your "Subgoals", then you can end the subproof and you may be able to use that subproof with the appropriate rule.
    - If the new formula fulfills your main goal, then (assuming you're not still inside a subproof) you are done!

As I mentioned before, Natural Deduction is a **complete** proof system, meaning that any valid argument of propositional logic can be proven using the eight Natural Deduction rules. Not only that, but the vast majority of valid arguments can be proven using this strategy. You could even program a computer to write your proofs for you (although it's not a trivial task). Certainly, any Natural Deduction proof I ask you to write in this class (except maybe on a bonus problem) will be for a **constructive** argument, meaning that you can find a proof of it using the above strategy.

Non-Constructive Proofs and the Principle of Explosion

But not every valid argument can be proven using these strategies. There are a couple of tricks that go beyond this general strategy of only using rules motivated by the formulas you have and the formulas that you want. I'll show you one of those tricks right in a moment, but first, let's start by looking at a simple proof of a weird claim.

$$\begin{array}{l} A \\ \neg A \\ \hline A \wedge \neg A \end{array}$$

When it comes to proofs, this is about as easy as they come.

Claim. $A, \neg A \vdash B$

Proof.
Assume A and $\neg A$.
From A and $\neg A$, we can conclude $A \wedge \neg A$. ($\wedge$-Intro) □

Now if you're paying attention, you may have realized that this argument is a *trivial* argument, meaning that the set of premises is *consistent.* It's a totally useless argument because you'll never have a situation where A and $\neg A$ are true, no matter what A is supposed to represent. However, if you recall, we decided that trivial arguments are technically considered "valid". This allowed us to keep rules like "an argument can only be proven invalid by giving a truth assignment that satisfies the premises and not the conclusion."

The above proof is another reason why "valid" is the correct choice for trivial arguments. Every step in the proof is completely reasonable, and we wouldn't want to have to reject the proof just because there was something contradictory in the assumptions. (Imagine if the inconsistency in the assumptions *wasn't* obvious!)

Carrying this reasoning to its inevitable conclusion, it seems like we should be able to write a proof of *any* trivial claim, even ones like this:

Claim. $A, \neg A \vdash B$

After all, the corresponding argument is supposed to be considered valid:

$$\begin{array}{l} A \\ \neg A \\ \hline B \end{array}$$

There's no counterexample; there is no truth assignment that satisfies the premises but not the conclusion, so it must be valid. Now if Natural Deduction is a complete proof system, we should be able to write a proof that demonstrates the argument is valid.

And we can! But it's a strange little proof. We can certainly start the proof using the Constructive strategy, but we won't get very far.

Proof.
Assume A and $\neg A$.
$\vdots$

We have two formulas we can use (A and $\neg A$), and we have one main goal (B). Neither A nor B has any connectives. The formula $\neg A$ has the main connective $\neg$, but the $\neg$-Elimination rule (Double Negation) isn't applicable, since it's just a single $\neg$. So we're stuck.

Here's where we can use a non-constructive trick. The trick is to use a combination of Proof by Contradiction and Double Negation to try to prove B. It would look like this:

Proof.

Assume A and $\neg A$.

  Suppose towards a contradiction that $\neg B$.

  Note that we've already proven both A and $\neg A$.

Under the assumption $\neg B$, we had A and $\neg A$, which contradict each other, and therefore $\neg\neg B$ is true. (Contrad.)

Since $\neg\neg B$ is true, so is B. (Dbl. Neg.) □

This is a stupid proof, of course. But it's not stupid because it's wrong. It's just strange, particularly because we didn't even use the sub-assumption $\neg B$ to get our contradiction. We already had the contradiction. We could have proved absolutely any conclusion using the same trick!

But that actually makes sense. This is not a normal claim. It's a claim that is only *trivially* valid, so it shouldn't be too shocking that the proof hinges on a silly technicality.

This principle is important and interesting enough to get it's own name, the **Principle of Explosion**. Normally, the set of conclusions you can prove from a given set of assumptions is fairly limited. But if there's a contradiction in your assumptions, then the set of conclusions you can prove "explodes" to include absolutely every formula.

Fact (The Principle of Explosion)**.** If your assumptions are inconsistent or contradictory, then you can prove absolutely any conclusion.

Now this might seem like a bit of an oddity, but the same strategy can be used to prove less ridiculous (and non-trivial) claims, such as the following.

Claim (Disjunctive Syllogism)**.** $p \vee q, \neg p \vdash q$

Now the Constructive Strategy actually gets us quite a bit further than last time.

Proof.

Assume $p \vee q$ and $\neg p$.

  Case 1: Suppose p.

  $\vdots$

  Hopefully therefore q? (*some rule*)

  Case 2: Suppose q

  Nothing to show in this case!

Under either possibility p or q, we proved q, so q is true in general. (Cases) □

We're only stuck in Case 1, where we need to prove q and all we have to work with are the inconsistent formulas p and $\neg p$. But we've already seen that when you have contradictory formulas like this, you can prove anything!

Proof.

Assume $p \vee q$ and $\neg p$.

Case 1: Suppose p.

Assume $\neg q$ for the purposes of proof by contradiction.

We already have p and $\neg p$ from earlier assumptions.

We assumed $\neg q$, and we achieved a contradiction (p and $\neg p$), so $\neg \neg q$. (Contrad.)

From $\neg \neg q$, we can conclude q. (Dbl. Neg.)

Hopefully therefore q? (*some rule*)

Case 2: Suppose q

Nothing to show in this case!

Under either possibility p or q, we proved q, so q is true in general. (Cases) □

Most claims that might be asked to prove will not require any sneaky tricks like this. In particular, the strategy of using Proof by Contradiction and Double Negation to prove a positive claim should be used only as a last resort. Even when we get into more informal proofs, this stays true. Try to only use Proof by Contradiction when your goal is a negated statement unless you've exhausted all other options. So for example, Proof by Contradiction is a great choice if you are asked to prove that a number is irrational, or to prove that an algorithm will not run in polynomial time, but it would be a poor choice for trying to prove that a number *is* rational or for trying to prove that an algorithm *does* run in polynomial time.

1.14 Equivalence Proofs

The techniques of Natural Deduction are very useful, and anywhere you find proofs, you will find inference rules like Application, Direct Proof, Proof by Contradiction, and Proof by Cases. But there are some useful strategies that go beyond inference rules. One particularly useful technique is the **chain proof**.

Now you've almost certainly seen equality chains already, most likely in an algebra or trigonometry class. For example, you might have seen something like this:

$$\begin{aligned}(x-a)^2 &= (x-a)(x-a)\\ &= x^2 - ax - ax + (-a)^2\\ &= x^2 - 2ax + a^2\end{aligned}$$

This form of proof takes advantage of the fact that equality is **transitive**, meaning that if you know $x = y$ and $y = z$, then you can conclude $x = z$. So it makes sense to connect a bunch of numeric expressions together in a chain of equalities. In an example like this one, what we are really saying is something like:

> $(x-a)^2$ is the same thing as $(x-a)(x-a)$, which is the same thing as $x^2 - ax - ax + (-a)^2$, which is the same thing as $x^2 - 2ax + a^2$.

Of course, the only fact we usually care about is the fact that the first expression $(x-a)^2$ is equal to the last expression $x^2 - 2ax + a^2$, but we use this chain of intermediary values (which are all equal to each other) to establish *why* the first expression is equal to the last one. If we want, we can even add explanations of why each chain of the equality holds:

$$\begin{aligned}
(x-a)^2 &= (x-a)(x-a) && \text{(Definition of Squaring)}\\
&= x^2 - ax - ax + (-a)^2 && \text{(Distributing)}\\
&= x^2 - 2ax + a^2 && \text{(Collecting Like Terms)}
\end{aligned}$$

Note that the explanation "Distributing" is not justifying the expression $x^2 - ax - ax + (-a)^2$. The expression $x^2 - ax - ax + (-a)^2$ is just a number and actually needs no justification. What *does* need justifying is the fact that the expression $x^2 - ax - ax + (-a)^2$ *is equal to* the previous expression $(x-a)(x-a)$. So even though we often write one expression on each line, with an equals sign on the left and a justification off to the right, keep in mind that the justification is actually there to explain why the *equals sign* is correct. It's a subtle point, but if you understand that, it will help you avoid making some potentially confusing notational mistakes.

We'll definitely be making use of chain equalities later on in this class, along with chain inequalities ($>$ and $\leq$ are also transitive relations), but right now, we're going to practice chain proofs by doing *equivalence* chains on formulas of propositional logic. In addition to learning about chain proofs in general, this will help solidify your understanding of propositional logic as well.

Let's start with another motivating example. Consider the following claim.

Claim. $(A \wedge B) \vee (C \wedge D) \equiv (D \wedge C) \vee (B \wedge A)$

I think that most students who look at this equivalence can easily see that these to formulas really are equivalent. If we were to try to prove this using truth tables, the table would have 16 rows and 10 columns. And unlike proving non-equivalence (which only needs one assignment to serve as a counterexample), proving equivalence requires investigating every single truth assignment, so we'd need to fill out all 160 entries in that table.

So truth tables aren't always a good choice. What about Natural Deduction proofs? You could prove this equivalence by first writing a proof of $(A \wedge B) \vee (C \wedge D) \vdash (D \wedge C) \vee (B \wedge A)$ and then writing a proof of $(D \wedge C) \vee (B \wedge A) \vdash (A \wedge B) \vee (C \wedge D)$. And that wouldn't be too bad. You'd need to use Proof by Cases to isolate the two parts of the $\vee$. And inside each case, you'd need to use $\wedge$-Elimination and $\wedge$-Introduction to reverse the order of the $\wedge$ formulas, followed by Weakening to put the $\vee$ back together. The proofs aren't *extremely* long, but they are kind of tedious and repetitive, and they don't much resemble the sort of reasoning that proof writers would normally use in a situation like this. One of the primary reasons we learned Natural Deduction is the fact that Natural Deduction proofs usually look a lot like real-world informal proofs. Just not in this situation.

So what does "normal" reasoning look like for a problem like this? If I stopped a random computer student on the street and asked them to explain why these two formulas are equivalent, they might say something along the lines of "It doesn't matter what order you put formulas together with $\wedge$ or $\vee$, and the only difference between the two formulas here is that we flipped the order of the big $\vee$ and also the order of the little $\wedge$'s."

We all understand that order of the formulas in an $\wedge$ or an $\vee$ formula doesn't matter. To be more technical about it, we can say that $\wedge$ and $\vee$ are **commutative** operations. The commutativity of $\wedge$and $\vee$ is a natural *consequence* of the Natural Deduction inference rules, so *technically* we don't *need* special commutativity rules, but it's such a basic concept that it would be silly to ignore it.

So in this section, we're going to treat the commutativity laws for $\wedge$ and $\vee$ (along with several more basic equivalence laws) as basic laws and use them to prove more complex equivalences using equivalence chains. Let's write an equivalence proof of our earlier claim right now.

Claim. $(A \wedge B) \vee (C \wedge D) \equiv (D \wedge C) \vee (B \wedge A)$

Proof.

$$\begin{aligned}
(A \wedge B) \vee (C \wedge D) &\equiv (B \wedge A) \vee (C \wedge D) && (\wedge\text{-comm.})\\
&\equiv (B \wedge A) \vee (D \wedge C) && (\wedge\text{-comm.})\\
&\equiv (D \wedge C) \vee (B \wedge A) && (\vee\text{-comm.})
\end{aligned}$$

□

One important thing to note about equivalence rules that makes them very different than the inference rules we've been using is that you can apply equivalence rules to *parts* of formulas, as we did in the $\wedge$-Commutativity steps in this proof. That doesn't always work for one-directional inference rules. But for chain proofs using equality, logical equivalence, or other kinds of "congruences", you don't need to match the rule to the entire formula. So, as in this example, you can use $\wedge$-Commutativity any place you see $\wedge$, not just when $\wedge$ is the main connective.

Guidelines for Semi-Formal Equivalence Proofs

Even though this isn't a direct Natural Deduction proof, we are still going to maintain the standards of "semi-formal" proofs.

- Always write your claim before your proof.

- Only use one rule per step.
- Always cite the logical rule you are using.
- Only use the rules we've talked about in class or that appear in the lecture notes.

So let's take a look at some of the rules that we'll be using. Let's start with some basic laws involving $\wedge$ and $\vee$.

Equivalence	Name
$p \wedge q \equiv q \wedge p$	Commutativity of $\wedge$
$p \vee q \equiv q \vee p$	Commutativity of $\vee$
$p \wedge (q \wedge r) \equiv (p \wedge q) \wedge r$	Associativity of $\wedge$
$p \vee (q \vee r) \equiv (p \vee q) \vee r$	Associativity of $\vee$
$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$ $(p \vee q) \wedge r \equiv (p \wedge r) \vee (q \wedge r)$	Distributivity of $\wedge$ over $\vee$
$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$ $(p \wedge q) \vee r \equiv (p \vee r) \wedge (q \vee r)$	Distributivity of $\vee$ over $\wedge$
$p \vee p \equiv p$ $p \wedge p \equiv p$	Idempotency
$p \wedge (q \vee p) \equiv p$	Absorption

Most of these should strike you as obviously true. That's good! Basic rules *should* seem obvious. Justifying a non-obvious statement with a sequence of obvious facts is sort of the whole point of writing proofs.

The distribution laws of $\wedge$ over $\vee$ (and vice versa) might look a bit complicated at first, but they work the same way as the distribution law of multiplication over addition. The only reason they look more complicated is because we can't take advantage of order of operations to simplify the notation. Here's what the multiplication-addition distribution law looks like when you have to use parentheses: $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$.

There is one important difference between how distribution works for propositional logic and how it works for arithmetic, and that's that the distribution works *both ways*. There's a distribution law for $\wedge$ over $\vee$ as well as a distribution law for $\vee$ over $\wedge$. Which is nice!

If you're interested in *why* these rules all hold and you're having a hard time holding the entirety of the distribution laws in your head at once, you can break them down into two implications, and try to prove them both using Natural Deduction: $p \wedge (q \vee r) \vdash (p \wedge q) \vee (p \wedge r)$ and $(p \wedge q) \vee (p \wedge r) \vdash p \wedge (q \vee r)$. And even if the rules do seem obvious to you, it makes for good practice.

The Idempotency and Absorption laws aren't super important, but they are occasionally necessary under specific circumstances. In particular, I wouldn't bother memorizing the names of those rules. You can always look them up when you need them.

Let's take a look at some rules involving negation.

Equivalence	Name
$p \equiv \neg\neg p$	Double Negation
$\neg(p \wedge q) \equiv \neg p \vee \neg q$ $\neg(p \vee q) \equiv \neg p \wedge \neg q$	De Morgan's Laws

This is the full equivalence rule version of Double Negation that can be used to eliminate a double negation anywhere in a formula or to *introduce* a new double negation anywhere into a formula.

Take a close look at De Morgan's Laws, and make sure you understand why they are true. Negation does *not* distribute over either $\wedge$ or $\vee$. And if you think about it carefully, you should be able to convince yourself of why this is the case.

For example, if you believe that $\neg(p \wedge q)$ is true, that means you believe that the claim "Both p and q are true," is *wrong*. An "and" claim like this is very strong, requiring both parts to be true for the whole thing to be true. It's actually quite easy to *disprove* such a claim: you only need to show that either p *or* q is false. You should also take a minute to think about the other version of De Morgan's Law, just to solidify it in your mind.

Most of the rules we'll use in this class involve the relationships between $\wedge$, $\vee$, and $\neg$. There are plenty of rules involving the other connectives, but the most useful ones are the ones that allow you translate those symbols into $\wedge$, $\vee$, and $\neg$, and these are the ones I'd like you to use in this class.

Equivalence	Name
$p \to q \equiv \neg p \vee q$	Implication / Material Implication
$p \leftrightarrow q \equiv (p \wedge q) \vee (\neg p \wedge \neg q)$	Bi-Implication
$p \oplus q \equiv (p \wedge \neg q) \vee (\neg p \wedge q)$	Exclusive Disjunction

This is enough rules to prove a whole host of interesting equivalences, such as the fact that any implication is equivalent to its contrapositive.

Claim. $A \to B \equiv \neg B \to \neg A$

Proof.

$$\begin{aligned} A \to B &\equiv \neg A \vee B && \text{(Impl.)} \\ &\equiv B \vee \neg A && (\vee\text{-Comm.}) \\ &\equiv \cdots ? \end{aligned}$$

□

At this point, you might or might not see how to finish the problem. But it's useful to consider more general strategies that will work even if you find yourself stuck. My first general strategy is to translate *both* formulas so that the only connectives are $\wedge$, $\vee$, and $\neg$. But be careful, if you want to work on both formulas, you have to make sure that you don't mix them up too early. One way to do this is to maintain two separate proof chains, like this:

Proof.

$$\begin{aligned} A \to B &\equiv \neg A \vee B && \text{(Impl.)} \\ &\equiv B \vee \neg A && \text{($\vee$-Comm.)} \end{aligned}$$

$$\begin{aligned} \neg B \to \neg A &\equiv \neg\neg B \vee \neg A && \text{(Impl.)} \\ &\equiv B \vee \neg A && \text{(Dbl. Neg.)} \end{aligned}$$

□

And this is a perfectly acceptable proof! We have proven that $A \to B$ and $\neg B \to \neg A$ are both equivalent to $B \vee \neg A$, and so therefore they both must be equivalent to each other. If you really love the simplicity of a single-chain proof, you can combine the two chains together (reversing the order of the second chain):

Proof.

$$\begin{aligned} A \to B &\equiv \neg A \vee B && \text{(Impl.)} \\ &\equiv B \vee \neg A && \text{($\vee$-Comm.)} \\ &\equiv \neg\neg B \vee \neg A && \text{(Dbl. Neg.)} \\ &\equiv \neg B \to \neg A && \text{(Impl.)} \end{aligned}$$

□

This single-chain proof would actually be pretty tricky to find all in one go as it requires a seemingly random insertion of a double negation, but it's actually pretty easy if you work both sides towards the middle.

Notational Abuses in Equivalence Proofs

Unfortunately, this desire to work both sides towards the middle often leads students to abuse the notation of chain proofs.

The following "proof" is an example of an abuse of notation. Never turn in a proof like this:

Not a Proof.

$$\begin{aligned} A \to B &\equiv \neg B \to \neg A && \\ \neg A \vee B &\equiv \neg\neg B \vee \neg A && \text{(Impl.)} \\ \neg A \vee B &\equiv \neg A \vee B && \text{(Dbl. Neg.)} \end{aligned}$$

I repeat: This is not an acceptable proof!

I see "proofs" like this every semester, and they always get 0 points and lots of grumpy students have to resubmit their assigments. And I do understand the thinking behind a "proof" like this. It's meant to mean something like this:

> I want to show that $A \to B \equiv \neg B \to \neg A$. Well, in order to prove that, I need to show that $\neg A \vee B \equiv \neg\neg B \vee \neg A$ (which is the same statement because of the Implication rule). And in order to show that, I need to prove $\neg A \vee B \equiv \neg A \vee B$ (which is the same as the previous statement because of Double Negation). But that statement is immediately obvious!

This is sound reasoning, but this isn't what the "proof" actually says as written. There are no comments about "wanting to show" something. Instead, this "proof" just starts off with a flat declaration $A \to B \equiv \neg B \to \neg A$ with no justification whatsoever. It then proceeds to use this unjustified claim to prove another equivalence and then finally to prove something ($\neg A \vee B \equiv \neg A \vee B$) that we already knew was true. In other words, this proof is *backwards*. It starts with the claim and then uses that claim to prove something obvious.

But that's now how proof is supposed to work. We don't assume our goal and then use that to prove something obvious. We're supposed to start with obvious facts, and then use those to prove the claim.

So here is how to avoid getting your problems rejected: NEVER START A PROOF BY DECLARING THAT YOUR GOAL IS TRUE. (Yes, you should write the claim *before* the proof, but that's not part of the proof itself.) Instead, the first line of your proof should begin with *one side* of the desired equivalence and a statement that it is equivalent to some other formula *according to some rule*. Every time you write down $\equiv$ in a proof, you should be able to justify with a rule why that equivalence is true.

Okay, let's do one last example of a more complex equivalence.

Claim. $P \to (Q \to R) \equiv (P \wedge Q) \to R$

Proof.

$$\begin{aligned} P \to (Q \to R) &\equiv \neg P \vee (Q \to R) && \text{(Impl.)} \\ &\equiv \neg P \vee (\neg Q \vee R) && \text{(Impl.)} \\ &\equiv \cdots ? \end{aligned}$$

$$\begin{aligned} (P \wedge Q) \to R &\equiv \neg(P \wedge Q) \vee R && \text{(Impl.)} \\ &\equiv \cdots ? \end{aligned}$$

□

Following my get-rid-of-the-→'s strategy is a good fist step. Here's another good strategy: "push" all the negations "inward" as far as you can using De Morgan's Laws.

Proof.

$$\begin{aligned}
P \to (Q \to R) &\equiv \neg P \lor (Q \to R) && \text{(Impl.)}\\
&\equiv \neg P \lor (\neg Q \lor R) && \text{(Impl.)}
\end{aligned}$$

$$\begin{aligned}
(P \land Q) \to R &\equiv \neg(P \land Q) \lor R && \text{(Impl.)}\\
&\equiv (\neg P \lor \neg Q) \lor R && \text{(DeM.)}\\
&\equiv \neg P \lor (\neg Q \lor R) && \text{(DeM.)}
\end{aligned}$$

□

A Few Extra Rules

If you're not in the honors section, you won't be required to use the rules in this section, but you may find the last example proof useful anyway.

If we want to have a truly complete set of rules, allowing us to prove any logical equivalence, we'll actually need to slightly stretch our notion of what counts as a "formula" of propositional logic. In addition to having variables and connectives, we also want to allow two special constant formulas: $\top$ (a formula which is always true) and $\bot$ (a formula which is always false).

It's a pretty simple extension, and it gives us the extra language we need to express a few more useful rules.

Equivalence	Name
$p \land \neg p \equiv \bot$	Negation / $\land$ Inverse / Contradiction
$p \lor \neg p \equiv \top$	Negation / $\lor$ Inverse / Excluded Middle
$p \land \top \equiv p \equiv \top \land p$	$\land$ Identity
$p \lor \bot \equiv p \equiv \bot \lor p$	$\lor$ Identity
$p \land \bot \equiv \bot \equiv \bot \land p$ $p \lor \top \equiv \top \equiv \top \lor p$	Domination

These rules can be surprisingly useful, such as with the following claim.

Claim. $\neg(X \leftrightarrow Y) \equiv X \leftrightarrow \neg Y$

Proof.

$$\begin{aligned}
\neg(X \leftrightarrow Y) &\equiv \neg\big((X \land Y) \lor (\neg X \land \neg Y)\big) && \text{(Bi-Impl.)}\\
&\equiv \neg(X \land Y) \land \neg(\neg X \land \neg Y) && \text{(De Mor.)}\\
&\equiv (\neg X \lor \neg Y) \land (\neg \neg X \lor \neg \neg Y) && \text{(De Mor.)}\\
&\equiv (\neg X \lor \neg Y) \land (X \lor Y) && \text{(Dbl. Neg.)}\\
&\equiv \cdots ?
\end{aligned}$$

$$\begin{aligned} X \leftrightarrow \neg Y &\equiv (X \wedge \neg Y) \vee (\neg X \wedge \neg\neg Y) && \text{(Bi-Impl.)} \\ &\equiv (X \wedge \neg Y) \vee (\neg X \wedge Y) && \text{(Dbl. Neg.)} \\ &\equiv \cdots ? \end{aligned}$$

□

We can get pretty far using the two strategies I mentioned before. But after that, we have two very different looking formulas. The first formula has $\wedge$ as the main connective, with $\vee$ as a secondary connective. The second is the other way around, with $\vee$ as the main connective and $\wedge$ as the secondary connective.[25] If we can swap the main connective and secondary connective for one of those formulas, it will be much easier to line up the rest.

How can we do this? If you look at the rules we have, there's one major rule that does exactly the job we're looking for: distribution! In this case, we're going to need a couple steps of distribution, and things are going to get a bit hairy before they get better, but they will get better in this case!

Proof.

$$\begin{aligned} \neg(X \leftrightarrow Y) &\equiv \neg\big((X \wedge Y) \vee (\neg X \wedge \neg Y)\big) && \text{(Bi-Imp.)} \\ &\equiv \neg(X \wedge Y) \wedge \neg(\neg X \wedge \neg Y) && \text{(De Mor.)} \\ &\equiv (\neg X \vee \neg Y) \wedge (\neg\neg X \vee \neg\neg Y) && \text{(De Mor.)} \\ &\equiv (\neg X \vee \neg Y) \wedge (X \vee Y) && \text{(Dbl } \neg\text{)} \\ &\equiv \big((\neg X \vee \neg Y) \wedge X\big) \vee \big((\neg X \vee \neg Y) \wedge Y\big) && \text{(Distr.)} \\ &\equiv \big((\neg X \wedge X) \vee (\neg Y \wedge X)\big) \vee \big((\neg X \wedge Y) \vee (\neg Y \wedge Y)\big) && \text{(Distr.)} \\ &\equiv \cdots ? \end{aligned}$$

$$\begin{aligned} X \leftrightarrow \neg Y &\equiv (X \wedge \neg Y) \vee (\neg X \wedge \neg\neg Y) && \text{(Bi-Imp.)} \\ &\equiv (X \wedge \neg Y) \vee (\neg X \wedge Y) && \text{(Dbl } \neg\text{)} \\ &\equiv \cdots ? \end{aligned}$$

□

And here's where the new rules involving $\top$ and $\bot$ become useful.

[25] If you've studied Artificial Intelligence, you may recognize these two formats as **conjunctive normal form** (CNF) and **disjunctive normal form** (DNF).

Proof.

$$\begin{aligned}
\neg(X \leftrightarrow Y) &\equiv \neg\big((X \wedge Y) \vee (\neg X \wedge \neg Y)\big) && \text{(Bi-Imp.)}\\
&\equiv \neg(X \wedge Y) \wedge \neg(\neg X \wedge \neg Y) && \text{(De Mor.)}\\
&\equiv (\neg X \vee \neg Y) \wedge (\neg\neg X \vee \neg\neg Y) && \text{(De Mor.)}\\
&\equiv (\neg X \vee \neg Y) \wedge (X \vee Y) && \text{(Dbl } \neg\text{)}\\
&\equiv \big((\neg X \vee \neg Y) \wedge X\big) \vee \big((\neg X \vee \neg Y) \wedge Y\big) && \text{(Distr.)}\\
&\equiv \big((\neg X \wedge X) \vee (\neg Y \wedge X)\big) \vee \big((\neg X \wedge Y) \vee (\neg Y \wedge Y)\big) && \text{(Distr.)}\\
&\equiv \big(\bot \vee (\neg Y \wedge X)\big) \vee \big((\neg X \wedge Y) \vee \bot\big) && \text{(}\wedge\text{ Inv.)}\\
&\equiv (\neg Y \wedge X) \vee (\neg X \wedge Y) && \text{(}\vee\text{ Ident.)}
\end{aligned}$$

$$\begin{aligned}
X \leftrightarrow \neg Y &\equiv (X \wedge \neg Y) \vee (\neg X \wedge \neg\neg Y) && \text{(Bi-Imp.)}\\
&\equiv (X \wedge \neg Y) \vee (\neg X \wedge Y) && \text{(Dbl } \neg\text{)}\\
&\equiv (\neg Y \wedge X) \vee (\neg X \wedge Y) && \text{(}\wedge\text{-Comm.)}
\end{aligned}$$

□

To summarize the strategies we've used for proving equivalences of propositional logic:

1. Translate everything into ∧, ∨, and ¬.
2. Use De Morgan's Laws to push the negations all the way in.
3. Use Distributivity Laws to get the main connectives to match. Try for either ∧ as a main connective with ∨ as a secondary connective or vice versa.

These are far from the only strategies you'll need, but they should give you a good start.

2 Set Theory

We've already been using sets informally (sets of formulas, in particular), but now it's time to talk about them in more detail. The basic ideas are very simple, but there are a few concepts that are easy to mix up if you're not careful. So even if you've seen sets before, I recommend pretending that you're learning them for the first time. It will help solidify what you do know, and if you have any bad habits or misconceptions, you might be able to unlearn them.

In it's most abstract form, a **set** is a mathematical object that has **members**. (It's also common to use the word "element" instead of "member".) Pretty much anything you can formally describe can be a member of a set. We've already seen sets whose members are formulas, and in other courses, you've probably talked about sets whose members are numbers. But you can also have sets whose members are text strings, functions, computer programs, boolean values, geometric shapes, or even other sets.[26]

A set is often described as a collection that is "made up" of its members. In some ways, this is a good way to think about it because it reinforces the idea that a set is defined only by what its members are. If "two" sets have the exact same members, then they're really the same set (and not really "two" sets at all).

You can describe the set by listing out its members, but to make it clear that you're talking about a set (and not a list or some other kind of collection), we put a pair of {curly braces} around the list. This notation is called **set list notation**.

For example, the set that contains the formulas $P \to Q$, $\neg R$, and $\neg P \land R$ could be written $\{P \to Q, \neg R, \neg P \land R\}$. It could also be written $\{\neg R, P \to Q, \neg P \land R\}$. These are the exact same set. It does not matter what order you list the members in the set; *it only matters what those members are.* It doesn't even matter if you list the members more than once (although there's usually not much point in doing so). The set $\{\neg R, P \to Q, \neg R, \neg P \land R\}$ is still the same set; it's just written in a different (and silly) way. Because there's more than one way to write a set, we use an equal sign to indicate that two sets are the same, as in $\{\frac{1}{1}, \frac{1}{2}, \frac{2}{2}, \frac{1}{3}, \frac{2}{3}, \frac{3}{3}, \frac{1}{4}, \frac{2}{4}, \frac{3}{4}, \frac{4}{4}\} = \{\frac{1}{4}, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}, \frac{3}{4}, 1\}$. (Do you see why these are the same set?)

Note: *the braces are important.* If you write your set using (parentheses), using [square brackets], using ⟨angled brackets⟩, or not using anything at all, you're going to confuse people. Each of these other symbols has other special meanings (we'll see some of them later), and if you use them, you're not talking about sets at all. If you leave off the braces entirely, things will quickly get very confusing, especially when the members of a set are other sets.

Question. Is the following statement true or false?
$\big\{\{1\}, \{1,2\}, \{1,2,3\}\big\} = \big\{1,1,2,1,2,3\big\}$

Answer. False! The set $\{1\}$ is a member of the left hand side, but not a member of the right hand side.

In more detail, the set on the right-hand side has three members (the numbers 1, 2, and 3), while the set on the left-hand side has three different members (the sets $\{1\}$, $\{1,2\}$, and $\{1,2,3\}$). Note that the number 1 and the set $\{1\}$ are *not* the same thing at all.

[26] When you're learning about sets, you'll sometimes see toy examples the members of our sets are vaguely defined things like people, types of animals, physical objects, spoken languages, or other silly things, but in reality, the members of sets really should be formally defined things. Otherwise, you can get bogged down with fuzzy questions like "Do neanderthals count as 'people'?" or "Are birds dinosaurs?" or "Does each grain of sand in a pile of sand count as a separate object?" But sometimes it's useful to look at familiar objects, so such examples aren't completely without merit.

In my answer above, I pointed out one thing that is a member of one of the sets and not a member of the other. There were many other options too (e.g., 2 is a member of the right hand side, but not the left; $\{1, 2\}$ is a member of the left hand side, but not the right; etc.). But **to prove that two sets are not equal, you only need to give an example of a member of one set that is not a member of the other set.** To show that two sets *are* equal, you'd have to show that every member of the first set is also a member of the second set and vice versa (either by exhaustively listing them and checking everything, or by some sort of proof). The claim that two sets are equal is a *universal* claim, so this shouldn't be too surprising.

When trying to justify a claim like this, many students end up working too hard, giving a broad explanation for why the two sets are different, such as "The set on the left is a set of sets, while the set on the right is a set of numbers." This is not an incorrect statement, but it's working too hard. If you can get away with a counterexample, you should give a counterexample; you don't have to explain why there are lots of counterexamples.

To make them easier to talk about, we sometimes assign a variable to use as a name for a set, as in $A = \{1, 2, 3\}$. You'll notice that most people use uppercase letters (e.g., A, S, X, Γ) for the names of sets and lowercase letters (e.g., a, s, x, γ) for the names of members of those sets. Math and logic are case sensitive! You might want to make sure you've got a way of writing your uppercase and lowercase letters differently[27], so that people can tell them apart.

When doing examples for this class, I might write something like $B = \{a, b, c\}$ and then never tell you what a, b, and c are. In a case like this, you can assume that a, b, and c are just letters of the English alphabet and not variables. The reason I can get away with this is because for some of our purposes, we often don't care what sorts of things those members are. I really just want to talk about a set with three members, and even though I don't care what the members are, I do need names to tell them apart.

I'm going to make some definitions now, so that I can use them later in the examples.

$$\begin{aligned} A &= \{1, 2, 3\} \\ B &= \{0, 1, 2, 3\} \\ M &= \{a, b, c\} \\ X &= \big\{\{1\}, \{1, 2\}, \{1, 2, 3\}\big\} \end{aligned}$$

Everything that can be said about sets can be brought back to the concept of membership. Even deciding whether two sets are equal is really just a case of checking to make sure the members are the same. So of course, we have a special symbol $\in$ to say that some thing is a member of some set. (LaTeX:

[27] I often use extra serifs on uppercase letters like C, S, X, and Z to tell them apart from the smaller c, s, x, and z. Or I'll use cursive on the lowercase letters (especially l).

\in, Unicode: U+2208, HTML: &isin;) This symbol is a stylized version of a lowercase Greek epsilon. A lowercase epsilon normally looks like this: ε, but that symbol has other meanings (especially for theoretical computer science), so make sure you use $\in$ and not ε to describe membership.

So to say that 2 is a member of A, we write $2 \in A$. When people read this out loud, sometimes they get lazy and say "two is *in* A", but this can be very confusing, especially when it comes to sets like X. Does it make sense to say that 2 is "in" X? It's certainly not a *member* of X. It *is* a member of $\{1, 2\}$ and $\{1, 2\}$ is a member of X, but that's not the same thing at all.

Example 2.1. Answer the following questions.

(a) Is $0 \in B$?

(b) Is $\frac{6}{3} \in A$?

(c) Is $a \in M$?

(d) Is $a \in A$?

(e) Is $A = B$? Why or why not?

(f) Is $1 = \{1\}$?

(g) Is $\{1, 2, 3\} \in X$?

(h) Is $\{1, 2, 3\} \in A$?

(i) Is $\{1\} \in X$?

(j) Is $1 \in X$?

Answer. Some of these are easy, but that doesn't mean they're not important.

(a) Yes.

(b) Yes, because $\frac{6}{3}$ is just another way of writing 2.

(c) Yes.

(d) No.

(e) No, because $0 \in B$, but $0 \notin A$.

(f) No. 1 is a number, and $\{1\}$ is a set. (That set happens to contain 1 as its only member, but it's still a set and not a number.)

(g) Yes.

(h) No. While all of the members of $\{1, 2, 3\}$ are members of A, the set itself is *not* a member of A.

(i) Yes.

(j) No. Remember $1 \neq \{1\}$.

The $\in$ symbol is a funny one. It's a relation symbol like $=$, $\geq$, $\equiv$, or $\vdash$, meaning that when you use it between two (of the right kind of) things, you're making a statement that's either true or false. (It's like a verb in English.) But it's a little funny because the things you use it to relate are not necessarily the same kind of thing. We use $\geq$ to compare two real numbers (when the left is at least as big as the right) and we use $\equiv$ to compare two formulas (when they are logically equivalent). We use $\in$ to compare some thing on the left (which could be anything, really) to some set (and it *must* be a set) on the right.

2.1 Set-builder Notation

Describing sets by listing their members is useful, but sets don't really become important until we give ourselves ways to describe them that don't require listing all the members out (which is tedious at best, and sometimes even impossible). This is where **set-builder** notation comes into practice. Anyone[28] who has anything to do with math, logic, or theoretical computer science needs to be able to use set-builder notation without thinking hard.

Remember when we defined our set $B = \{0, 1, 2, 3\}$? You could describe this as "the set of all integers between 0 and 3, including 0 and 3". It's long-winded in English, but sometimes even a messy English description is better than listing the members. What if I wanted the set C to be the set of all integers between 0 and 100, including 0 and 100?

I could write $C = \{0, 1, 2, 3, \ldots, 99, 100\}$, and that works, but only if the pattern implied by the dots is really obvious. Never use $\ldots$ unless you're sure that the pattern is clear to the reader. So for example $N = \{0, 1, 2, 3 \ldots\}$ is fine and so is $T = \{0, 3, 6, 9, 12, \ldots\}$, and you can probably get away with something like $S = \{0, 1, 4, 9, 16 \ldots\}$, but it doesn't take much for a fairly simple pattern to become difficult to read. For example, what if I were to give a "definition" like $Q = \{3, 4, 6, 8, 12, 14, 18, \ldots\}$. That's hardly an obvious pattern, and yet it's not a very complicated set

Is it obvious to you what the next number in that list is? Some of you may be able to figure it out, but it's not at all obvious, which means that we really shouldn't use $\ldots$ to describe it. For the record, Q is supposed to be the set of all numbers that are one larger than a prime number.

In these cases, to communicate this concept of "the set of ____'s that are ____," we use set-builder notation. Set-builder notation surrounds the description of the set with curly braces just like before, but instead of listing the members, we give a rule for how to generate members of the set. The rule is broken into two parts, which are separated by a vertical bar $|$ (or sometimes a colon :). To the left of the bar is a sort of template representing a generic member of the set. That template should have at least one undefined variable in it (it will get defined by the other part of the notation). To the right of the

[28] Not everyone calls it "set-builder" notation, but everyone knows how to use it.

bar is a statement about the variable(s) on the left which establishes what kinds of values they can take.

So we could rewrite Q like this: $Q = \{p+1 \mid p \text{ is a prime number}\}$. I would read this out loud as "Q is the set of all numbers of the form $p+1$, such that p is a prime number," or maybe "Q is the set of all $p+1$, where p is prime." So for example, 3 is a member because it can be written as $3 = 2+1$ and 2 is a prime number. Similarly, $258 \in Q$ because $258 = 257+1$ and 257 is prime.

Along the same lines we could also rewrite our definition of the set S as $S = \{n^2 \mid n \text{ is an integer }\}$. And if we want to make our definitions even more compact, we can make use of a few common abbreviations for some useful sets of numbers.

For example, the set of **natural numbers** $\{0, 1, 2, 3, \ldots\}$ is written $\mathbb{N}$. That funny double-line notation is called "blackboard bold" because it used to be (and still is for some people) that the natural numbers was typeset using ordinary bold **N**, but that was too hard to write on a blackboard. Another common set is the set of all **integers** $\{\ldots - 3, -2, -1, 0, 1, 2, 3, \ldots\}$, which is written with a blackboard-bold capital $\mathbb{Z}$.[29] We'll see some more important sets of numbers later.

Since we have a symbol for the set of integers, we could actually shorten this quite a bit, down to: $S = \{n^2 \mid n \in \mathbb{Z}\}$. Note that there's often more than one way to represent a set using set-builder notation. So I could also write $S = \{s \mid n \text{ is a perfect square }\}$.

Let's take a stab at writing $B = \{0, 1, 2, 3\}$ in set-builder notation. There's always more than one way, but there's often an obvious way. In this case, I need to communicate that all the members are integers, and that they are all greater than or equal to 0 and less than or equal to 3. So I could write $B = \{n \mid n \in \mathbb{Z} \wedge n \geq 0 \wedge n \leq 3\}$. Feel free to use any of the logical connectives to combine several requirements into one requirement. Most of the time, you'll just be using $\wedge$ and maybe $\neg$, but occasionally $\vee$ will be useful.[30]

Here are a few example problems to help get used to things.

Example 2.2. Decide if the following statements are true or false, and explain why or why not.

(a) $3 \in \{x \mid x \in \mathbb{Z} \wedge -x \geq -5\}$

(b) $6 \in \{mn \mid m \in \mathbb{N} \wedge n \in \mathbb{N}\}$

(c) $-6 \in \{mn \mid m \in \mathbb{N} \wedge n \in \mathbb{N}\}$

(d) $A \wedge A \in \{p \mid p \text{ is a tautology in propositional logic}\}$

(e) $\neg A \in \{p \mid p \text{ is a propositional logic formula} \wedge \neg p \equiv A\}$

(f) $\{0, 1\} \in \{T \mid 2 \in T\}$

[29] The letter "Z" is used here because the German word for number is "zahl."

[30] You *can* use $\rightarrow$, but it usually just makes things confusing.

(g) $\{2\} \in \{T \mid 2 \in T\}$

(h) $2 \in \{T \mid 2 \in T\}$

(i) $\mathbb{N} \in \{T \mid 2 \in T\}$

Answer. In every example, finding the answer is just a matter of trying to match the potential member (on the left of the $\in$) to the expression (to the left of the $|$) in such a way that the proposition (to the right of the $|$) is true.

(a) True. When $x = 3$, $-x = -3 \geq -5$.

(b) True. One way to match 6 to mn is to make $m = 2$ and $n = 3$, then $6 = mn$ and both m and n are members of $\mathbb{N}$.

(c) False. Since m and n both have to be natural numbers, they are nonnegative, and so mn must also be nonnegative. Since -6 is negative, it can't be written mn for natural numbers m and n.

(d) False. $A \wedge A$ is not a tautology.

(e) True. When $p = \neg A$, $\neg p = \neg\neg A$, and $\neg\neg A \equiv A$.

(f) False. $2 \notin \{0, 1\}$

(g) True. $2 \in \{2\}$

(h) False. 2 is not even a set, so it has no members.

(i) True. $2 \in \mathbb{N}$

This is as good a place as any to introduce some more standard notations. Previously, I defined the symbols $\mathbb{N} = \{0, 1, 2, 3, \ldots\}$ (the natural numbers) and $\mathbb{Z} = \{\ldots, -3, -2, -1, 0, 1, 2, 3, \ldots\}$ (the integers). There are a couple other commonly used sets of numbers whose names and symbols you should know. The first is the set of **rational numbers**. If you have trouble remembering which numbers are rational numbers, keep in mind that it's actually **ratio**nal numbers, as in numbers which can be ratios (fractions) of integers. The symbol that is used is a bold or blackboard bold capital $\mathbb{Q}$, which stands for "quotient" (which is really just another word for ratio). (It's not $\mathbb{R}$ because we'll need that for the next set. Keep reading.) $\mathbb{Q}$ obviously includes numbers like $\frac{1}{2}$ and $\frac{5}{27}$, which are most naturally written as ratios, but it also includes numbers like $4\frac{2}{3}$ and 7 because while I didn't write those as ratios just now, they certainly *can* be written as a ratios: $4\frac{2}{3} = \frac{16}{3}$ and $7 = \frac{7}{1}$.

A side note about handwriting blackboard bold letters. Most blackboard bold letters can be formed by just writing the letter and adding an extra vertical line to them. This doesn't end up looking *exactly* like the versions that are printed, but that's okay. So to handwrite a blackboard bold $\mathbb{N}$, just draw a regular capital "N" and then add an extra vertical line on the left side. It usually comes out looking more like **IN** than $\mathbb{N}$ when you handwrite it. The same goes for $\mathbb{Q}$, $\mathbb{R}$, and $\mathbb{C}$, which usually come out looking more like $\mathbb{Q}$, $\mathbb{R}$, and $\mathbb{C}$ when you write them by hand. There's a bit of a trick with $\mathbb{Z}$. Instead of writing a "Z" and then trying to add a line, which sort-of-but-not-really works, but there's a better way. First write a "7", and then add a slightly-angled "L": $\angle$. When you put them on top of each other, you'll end up with a very satisfying $\mathbb{Z}$.

Set-builder notation is a natural way to describe sets whose members "can be written" in some particular form. We can define $\mathbb{Q}$ very succinctly using this method: $\mathbb{Q} = \{\frac{p}{q} \mid p \in \mathbb{Z} \wedge q \in \mathbb{Z} \wedge q \neq 0\}$. It looks monstrous, but if you break it down, it's not so bad. It's surrounded with braces, so it's a set. Which set is it? Well, it's the set of things of the form $\frac{p}{q}$ where p and q are integers and q is not zero. Why do we have to specify that $q \neq 0$? It's just a technicality to avoid saying that undefined "things" like $\frac{2}{0}$ are rational numbers.

Example 2.3. Decide if the following numbers are rational.

(a) $-\frac{7}{4}$

(b) $\frac{10}{4}$

(c) $\frac{24}{8}$

(d) 3

(e) 0.25

(f) $0.\overline{3}$

(g) π

(h) $\sqrt{5}$

(i) $\sqrt{9}$

(j) $\frac{pi}{2}$

(k) $\frac{0.5}{0.25}$

(l) $\sqrt{-4}$

Answer. We can test all these just by looking to see whether we can write them in the form $\frac{p}{q}$ using the right kind of p and q. If we can, we should be able to say what p and q are.

(a) Yes! $p = 7$ and $q = 4$. Both of those are integers and $4 \neq 0$.

(b) Yes! $p = 10$ and $q = 4$.

(c) Yes! $p = 24$ and $q = 8$.

(d) Yes! While it's not written as a fraction *right now*, it *can* be written in that way. One way to do it would be to write $3 = \frac{24}{8}$, which we already said was a member of $\mathbb{Q}$. (Of course, an easier way would be to write it $\frac{3}{1}$. Using this technique, we can write *any* integer in this form, implying that *every integer is also a rational number*.)

(e) Yes! It can be written in the form $\frac{1}{4}$, so $p = 1$ and $q = 4$. (Of course, it's even easier to write it as $\frac{25}{100}$, we do read the number as "twenty-five hundredths" after all. This trick works for *any* terminating decimal.)

(f) Yes! $0.\overline{3}$ can be rewritten as $\frac{1}{3}$. (It's not obvious, but you may remember from high school algebra that *any* repeating decimal can be written as a ratio of integers. Another non-obvious fact that you should remember is that if the decimal expansion of a number does not repeat and never ends, then it is *not* rational.)

(g) No. I wouldn't ask you to prove this in this class. It is *not* an obvious thing and actually quite difficult to prove. However, π is famous for not being rational, so even if you don't know why it's not rational, you should know *that* it is irrational.

(h) No. The square, cube, nth roots that don't "come out nicely"[31] are never rational numbers. You don't need to know why this is true, but again, you should know that it is true.

(i) Yes! This one's tricky, but only because I wrote the number in a funny way. Another way to write $\sqrt{9}$ is 3, and another way to write that is $\frac{3}{1}$, which shows us that it is a rational number.

(j) No! While we can write it as a ratio, we cannot write it as a ratio of integers.

(k) Yes! While the most obvious way to write this as a ratio doesn't use integers, we *can* write it in the form $\frac{1}{2}$, which does use integers. My point here being that it's not a question of how it's currently written, but a question of how it *can* be written.

(l) No! While we can write $\sqrt{-4} = 2i = \frac{2i}{1}$, $2i$ is not an integer. Heck, it's not even a real number.

I'm not going to go into defining them carefully, but you should also know that $\mathbb{R}$ is used as a symbol for the **real numbers**, which you can just think of as anything that can be written with a decimal expansion (even numbers with infinite decimal expansions with no patterns to them). Or alternately, you can think of a real number as anything that appears on the number line. All of the above examples are real numbers except for $\sqrt{-4}$, which is not real.

We won't use it in this class, but you should be aware that $\mathbb{C}$ is used as a symbol for the **complex numbers**, which includes the real numbers, along

[31] This is not very precise, I know, but this isn't the right class to make it any more precise.

with all the imaginary numbers and all the possible combinations of real and imaginary numbers.

2.2 The Empty Set

Consider the sets $A = \{x \mid x \in \mathbb{N} \wedge x \text{ is even } \wedge x \text{ is odd }\}$ and $B = \{x \mid x \in \mathbb{R} \wedge x < -2 \wedge x > 2\}$. How many members are in A? How many in B? In both cases, there is nothing that meets the requirement for being a member of the set, so neither set has any members. But remember that a set is *defined* by which things are members of it, so not having any members is the only defining feature of both sets. I guess that's just a long-winded way of saying that they're the same set: $A = B$. The set with no members at all is called the **empty set**, and while you can define it in a roundabout way like I did with A and B above, you can also define it using the list notation just by writing an empty pair of braces: $\{\}$. So $A = \{\} = B$. It crops up a lot, so it also has a special notation: $\varnothing$. It's important to think of $\varnothing$ as being shorthand for $\{\}$.

WARNING: $\varnothing$ is not the same thing as the number zero 0. Back in the early days of computers, to distinguish between the letter O and the number zero, many computer fonts would put a slash inside of the zero. This has caused people learning set theory no end of trouble. So let me say it again: 0 is a number and $\varnothing$ is a set that has no members.

WARNING: Another misconception is that $\varnothing$ is just a way of saying "nothing" or "this space intentionally left blank." For example, people see $\{\varnothing\}$ and wrongly think, "Ah, that means the set that contains nothing!" That's not true! The set $\{\varnothing\}$ is *not* empty. It has a member. Granted, it only has one member $\varnothing$, and $\varnothing$ is empty, but the set $\{\varnothing\}$ is not empty. It's similar to the way a box that has an empty box inside it is not empty: it has a box inside it!

2.3 Set Operations

I'm guessing that most of you have already seen the basic set operations of intersection, union, complementation, and maybe set difference. We'll be tackling them from a slightly different angle today, but all the intuitions you've already formed (using Venn diagrams, etc.) are probably still useful.

Let's start by defining intersection and union. If you have two sets A and B, then their **intersection** (written $A \cap B$) is the set of all things that are both members of A *and* members of B. Similarly, the **union** of A and B (written $A \cup B$) is the set of all things that are either members of A *or* members of B. (For example, if $A = \{1, 2, 3, 4, 5\}$ and $B = \{0, 1, 4, 9\}$, then $A \cap B = \{1, 4\}$ and $A \cup B = \{0, 1, 2, 3, 4, 5, 9\}$.)

It is not coincidental that the symbols for intersection and union look similar to the symbols for conjunction $\wedge$ and disjunction $\vee$. Notice that the only real difference between the two definitions is that intersection is defined using "both…and" while union is defined using "either…or".

We can make this connection more concrete if we rewrite the definitions using set-builder notation. So we can define the intersection of A and B as

$A \cap B = \{x \mid x \in A \wedge x \in B\}$, and we can define their union as $A \cup B = \{x \mid x \in A \vee B\}$. Think about what these new definitions mean, and convince yourself that they agree with the earlier definitions we gave.

Next, we're going to look at the complement of a set, but we have to be careful about it. The general idea is that we're often interested in the set of all things that are *not* in a particular set. But this only really makes sense if we've got a good notion of what "all things" means. This wasn't a problem with intersection and union because when we wanted to figure out what was in $A \cap B$ or $A \cup B$, we only have to look at the members of A and B. But in this case, we have to look *everywhere*, and that leads to trouble unless you've got a well-defined concept of "everywhere". There are two ways to fix this problem, and we'll look at both of them.

The first way is to only talk about complements of sets when we have a fixed **universal set** $\mathcal{U}$. For any given situation, the members of the universal set include everything we could possibly want to talk about as members of a set. This may change depending on the problem we're working with, but if we don't say what it is and it's not clear from context, you should assume that it includes everything you can possibly think of (numbers, sets, formulas, people, letters, words, strings, physical objects, etc.). When you have a universal set in place, we now know what "everything" means, and we make the following definition. The **complement** of a set A is the set (written $\overline{A}$)[32] of everything in $\mathcal{U}$ that is *not* an member of A. Or, in set-builder notation: $\overline{A} = \{x \mid x \notin A\}$. So if A and B are as above, and $\mathcal{U} = \{0, 1, 2, \ldots, 10\}$, then $\overline{A} = \{0, 6, 7, 8, 9, 10\}$ and $\overline{B} = \{2, 3, 5, 6, 7, 8, 10\}$. Of course, if $\mathcal{U} = \mathbb{N}$, then $\overline{A} = \{6, 7, 8, 9, \ldots\}$.

If we don't want to fix a universal set (and sometimes it's useful not to), then we can't talk about the complement of a set. But we can still use the same basic idea of looking at all members of one set that are not members of another set. This is the idea of set difference (or subtraction), and it's exactly like complementation, except that we're not forced to use some universal set as the first set. We can use any set we want. Here's the definition of set difference: the **difference** between A and B (written $A \setminus B$) is the set of all members of A that are not members of B. In set-builder notation: $A \setminus B = \{x \mid x \in A \wedge x \notin B\}$. (Set difference is also sometimes written as subtraction: $A - B$ instead of $A \setminus B$. The text book we used to use writes it this way. You can write it whichever way you want, but I prefer to write it with a backslash because it makes it clear that I'm working with sets and not numbers.) For example, $\{1, 2, 3, \ldots, 10\} \setminus A = \{6, 7, 8, 9, 10\}$. Note that it's not necessary for the set on the left to contain the set on the right as a subset. In other words, it's entirely possible for there to be members of the right set that aren't in the left set at all. The way set difference is defined, you just end up ignoring those members. For example $A \setminus B = \{2, 3, 5\}$ and $B \setminus A = \{0, 9\}$.

I thought I'd do a few quick examples here, just to make sure we're all on the same page. These are all atypical, but important examples, which have answers that make sense, but that you might not be expecting.

[32] Some books will write A' or A^{C} for the complement of A.

Example 2.4. Let our universal set be $\mathcal{U} = \{1, 2, 3, \ldots, 100\}$, and make the following set definitions: $A = \{1, 2, 3, 4, 5\}$, $B = \{2, 4, 6, 8, 10\}$, $C = \{1, 3, 5, 7, 9\}$, and $D = \{1, 2, 3, \ldots, 10\}$. Compute the following sets.

- $A \cap D$

  This is every member of A that is also an member of D. Just go through the members of A and see which are also members of D. You should get $A \cap D = \{1, 2, 3, 4, 5\}$. Yes, this is exactly the same as A. There is nothing wrong with $A \cap D$ being the same set as A, just like there's nothing wrong with $5 + 0$ being the same number as 5. In fact, this will always happen if all the members of A are also members of D.

- $B \cup D$

  This is everything that is either an member of B or an member of D, so just go through both sets and make sure you include everything. $B \cup D = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$. This is the same as D itself, and again, this will happen anytime all the members of B are also members of D

- $B \cap C$

  If you run through the members of B and write down those that are also members of C, you should end up writing nothing down. And that's just fine! There are no members of the set $B \cap C$, so we can just write $B \cap C = \{\}$. Of course, this set is better known as the empty set, so you can also write $B \cap C = \varnothing$. This happens here because there is nothing that is an member of both sets. A pair of such sets are said to be **disjoint** from one another, but you don't need to remember that for this class.

- $B \setminus C$

  This is every member of B that is *not* an member of C. We can compute it in essentially the same way that we computed the intersection: by running through the members of B and seeing which ones are not members of C. In this case, *every* member of B is not in C, so we just get everything in B. That is, $B \setminus C = \{2, 4, 6, 8, 10\}$, which is the same set as B itself. This always happens when B and C are disjoint.

- $A \setminus D$

  Running through the members of A, we find that every one of them is also an member of D, so none of them can be members of $A \setminus D$. It has no members, so $A \setminus D = \{\} = \varnothing$. This always happens when all the members of A are also members of D.

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

  You don't really have to comprehend the whole formula in one go. Break it up into pieces, just like you would when calculating something like $(3 + 8) \cdot 2$. First, we compute $A \cap B = \{2, 4\}$. Then we compute the union of this set with C: $(A \cap B) \cup C = \{1, 2, 3, 4, 5, 7, 9\}$.

- $\overline{B \cup C}$

  When we put a vertical bar over a compound formula like this, it means that we're taking the complement of the entire set $B \cup C$. It does *not* mean that we're unioning the complements of B and C (that would be written $\overline{B} \cup \overline{C}$. Again, break it up. $B \cup C\{1, 2, 3, \ldots, 10\}$, so $\overline{B \cup C} = \{11, 12, 13, \ldots, 100\}$.

- $(A \cup B) \setminus \overline{A \cap C}$

  You have to work inside out, so we work on the innermost parentheses first: $A \cup B = \{1, 2, 3, 4, 5, 6, 8, 10\}$ and $A \cap C = \{1, 3, 5\}$. Using this, we can find $\overline{A \cap C} = \{2, 4, 6, 7, 8, \ldots, 100\}$. And finally, we get $(A \cup B) \setminus \overline{A \cap C} = \{1, 3, 5\}$.

2.4 Cardinality

How many members are in the set $\{0, 1, 2, 3\}$? I hope everyone says 4. Let's get a tiny bit tougher. How many members are in the set $\{0, 1, 2, 3, \frac{0}{2}, \frac{1}{2}, \frac{2}{2}, \frac{3}{2}\}$? Well, $\frac{0}{2}$ is just another way of writing 0 and $\frac{2}{2}$ is just another way of writing 1, so there are actually only 6 members here. How many members are there in $S = \{\{1\}, \{2\}, \{1, 2\}\}$? There are three members. Remember that 1 is not the same thing as $\{1\}$ just as a folder that contains the file `1.txt` is not the same as the file `1.txt` itself (just like I not the same thing as the club that only has me as a member (and just like a a box with one cookie in it isn't the same thing as a cookie)).

Okay, now let's get really tricky. How many members are there in $\mathbb{N}$? So far, the answer to the question "how many members?" has always been a natural number, but because we have to deal with infinite sets, we have to expand our notion of the "size" of the set. The concept of the size of a set actually gets very complicated when you talk about infinite sets. Thankfully, we won't have to deal with those complications in this course. I only mention it because it explains why we use the funny name **cardinality** to describe the number of members in a set. There are other notions of size, but the distinctions only matter for infinite sets, and for our purposes, it's enough to say that the cardinality of an infinite set is infinite.[33]

We write $|X|$ to mean the cardinality of the set X. These are the same vertical bars that are used to represent the absolute value of a real number (amongst other things). So we might write $|S| = 3$ or $|\{0, 1, 2, 3\}| = 4$, and we might say that $|\mathbb{N}|$ is infinite.

Example 2.5. Calculate the following.

(a) $|\{\frac{1}{1}, \frac{1}{2}, \frac{2}{1}, \frac{2}{2}\}|$

(b) $|\{\{1\}, \{2\}, \{1, 2\}\}|$

[33] It's true to say that the cardinality of $\mathbb{N}$ is infinite and that the cardinality of $\mathbb{R}$ is infinite, but it turns out not to be the case that their cardinalities are equal. But for this class, it's enough to say that both sets are infinite.

(c) $|\{\{1\},\{2\},\{1,2\},\{2,1\}\}|$

(d) $|\{x \mid x \in \mathbb{Z} \land x < 3 \land x > -3\}|$

(e) $|\{2x \mid x \in \mathbb{Z} \land x < 3 \land x > -3\}|$

(f) $|\{x^2 \mid x \in \mathbb{Z} \land x < 3 \land x > -3\}|$

(g) $|\{x \mid x \in \mathbb{N} \land x < 100\}|$

(h) $|\{x^2 \mid x \in \mathbb{Z}\}|$

Answer. Note that these are all numbers, not sets. In most of these cases, the sets are small enough that we can actually list out all the members.

(a) $|\{\frac{1}{1},\frac{1}{2},\frac{2}{1},\frac{2}{2}\}| = 3$. It looks like 4, but remember that $\frac{2}{2}$ is the same as $\frac{1}{1}$, so there are really only 3 members. We just listed one of them twice.

(b) $|\{\{1\},\{2\},\{1,2\}\}| = 3$. The members of this set are $\{1\}$, $\{2\}$, and $\{1,2\}$. 1 and 2 are not members of the set. They are members of members of the set, but that's not really relevant.

(c) $|\{\{1\},\{2\},\{1,2\},\{2,1\}\}| = 3$. This is actually the exact same set as the last one because $\{1,2\} = \{2,1\}$.

(d) $|\{x \mid x \in \mathbb{Z} \land x < 3 \land x > -3\}| = 5$. The members of this set are -2, -1, 0, 1, and 2.

(e) $|\{2x \mid x \in \mathbb{Z} \land x < 3 \land x > -3\}| = 5$. The members of this set are $2 \cdot -2 = -4$, $2 \cdot -1 = -2$, $2 \cdot 0 = 0$, $2 \cdot 1 = 2$, and $2 \cdot 2 = 4$.

(f) $|\{x^2 \mid x \in \mathbb{Z} \land x < 3 \land x > -3\}| = 3$. Yes, there are five different possible values of x, but remember that the members of the set are the different possible values of x^2, and there are only three of those $(-2)^2 = 4$, $(-1)^2 = 1$, $0^2 = 0$, $1^2 = 1$, and $2^2 = 4$. Some of them just show up more than once.

(g) $|\{x \mid x \in \mathbb{N} \land x < 100\}| = 100$. Too many to list them all out, but we can start, figure out the pattern, figure out when/if it ends, and then figure out the number of members. It's important to take a careful look at the edges of the conditions because sometimes you can get tripped up by technicalities. In this case, there are two places you might get tripped up: 0 and 100. 0 *is* a member because $0 \in \mathbb{N}$, and 100 is *not* a member because $100 \not< 100$. So $\{x \mid x \in \mathbb{N} \land x < 100\} = \{0,1,2,3,\ldots,98,99\}$. If it was just 1 through 100, there would be 100 members, but since we want to include 0 and not include 100, there are $100-1+1 = 99+1 = 100$ members. The two technicalities cancel each other out here, so if you guessed 100, you might just've gotten lucky.

(h) $|\{x^2 \mid x \in \mathbb{Z}\}|$ is infinite. Make sure you actually convince yourself that there are an infinite number of members here, and not just that there are an infinite number of x's. If you want to be sure, try to construct a sequence of members of the set that you know isn't going to have any repeats and you know will never end. In this case, you could look at $1^2 = 1$, $2^2 = 4$, $3^2 = 9$, $4^2 = 16$, ... Notice that I'm not saying this list will cover *all* the members of the set. (In particular, I haven't looked at what happens when $x = 0$ or x is negative.) But all of these *are* members of the set, and this pattern will keep producing members of the set, and it will never repeat itself, so that's a pretty good reason to think the set is infinite.

Question. What is the cardinality of the empty set? $|\varnothing| = ?$

Answer. Since $\varnothing$ has no members, its cardinality is zero. So $|\varnothing| = 0$. (Note: $\varnothing$ is not *equal* to zero. It just has zero members.)

2.5 Subsets

Recall that the membership relation $\in$ is a kind of tiered relationship that allows us to relate two (usually) different kinds of objects. On the right: a set, and on the left: an member of that set. So we could say $1 \in \{1, 2\}$ or even $\{1, 2\} \in \big\{\{1\}, \{1, 2\}\big\}$, but we could *not* say $\{1, 2\} \in \{1, 2, 3\}$. There is indeed a connection between the sets $\{1, 2\}$ and $\{1, 2, 3\}$, but it is not that kind of connection.

How would we describe the connection between these sets? Notice that while the set $\{1, 2\}$ is "inside of" the set $\{1, 2, 3\}$ in a sense, both of the objects are sets, and both sets' members are just numbers, so it's not a case of one being an *member* of the other. It's more of a relationship *between their members*: every member of $\{1, 2\}$ is also an member of $\{1, 2, 3\}$. There is a name for this relation, and it's called the "subset" relation. We say that a set A is a **subset** of another set B if every member of A is also an member of B, and we write $A \subseteq B$. This symbol intentionally looks like $\leq$ because according to the definition, every set is a subset of itself. For example, every member of $\{1, 2, 3\}$ is also an member of $\{1, 2, 3\}$ (stupid, but true), so we can say $\{1, 2, 3\} \subseteq \{1, 2, 3\}$. You can also read $A \subseteq B$ as saying B is a **superset** of A, and you can, as expected, write $B \supseteq A$, which means the same thing as $A \subseteq B$.[34] When we want to say that A is a subset of B, but that they aren't the same set (this isn't very common, surprisingly), we say A is a **proper** subset of B.

Most textbooks will tell you that the word "contains" means "has as a subset," as in "the set $\{1, 2, 3\}$ contains $\{1, 2\}$, and occasionally you'll find people using "contains" this way, but that is *not* the only use of the word. It's not even the most common use of the word. For example, you will often hear $\{1, 2\}$ read out loud as "the set containing 1 and 2." Now that certainly

[34] Warning: many, many peoples use $\subset$ instead of $\subseteq$, and confusingly, it usually means the same thing: equality is acceptable, which is completely different than with $<$. To avoid confusion, I will always use $\subseteq$, and if I want to disallow equality, I will use $\subsetneq$.

doesn't mean that 1 and 2 are *subsets* of $\{1,2\}$ because that wouldn't make any sense. (They're not sets at all!) In this context, "contains" means "contains as a *member*," not "contains as a *subset*."

So which is correct? Does "contains" mean "has as a subset" or "has as a member"? The answer is that both are correct. "Contains" is an ambiguous word, like "in," and people depend on context to figure out which meaning is intended. In this class, I will try to avoid using the word "contains" like this, and you should avoid doing so too.

Example 2.6. Let $A = \{0,1,2,3\}$, $B = \{1,2,3\}$, $D = \{n \mid n \in \mathbb{Z} \wedge 0 < n < 4\}$, $S = \{n^2 \mid n \in \mathbb{Z}\}$, and $X = \big\{\varnothing, \{1\}, \{2\}, \{1,2\}\{1,2,3\}\big\}$. Answer true or false (with justification):

(a) $A \subseteq B$

No. Because $0 \in A$, but $0 \notin B$. In general, this is how you justify a claim that one set is not a subset of another, you give an example of an member of the claimed subset (left) that is not an member of the claimed superset (right).

(b) $B \subseteq A$

Yes. Each member (1, 2, and 3) of B is also an member of A. In general, it's harder to show that one subset *is* a subset of another, you have to show that *every* member of the subset is an member of the superset. In this case, B has three members, so there's only three things to show.

(c) $B \in A$

No. Just checking.

(d) $B \subsetneq A$

Yes. B is a subset of A as we showed earlier, and $A \neq B$ because $0 \in A$, but $0 \notin B$.

(e) $\{2\} \subseteq A$

Yes. We need to show that every member of $\{2\}$ is an member of A. In this case, there is only one member of $\{2\}$, 2, and 2 is an member of A.

(f) $B \subseteq D$

Yes. There are three members of B, so we will need to show that all three of them are members of D. Because we defined D with set-builder notation, showing that something is in D involves first, writing it in the form n and second, showing that this n satisfies the two conditions (that $n \in \mathbb{Z}$ and that $0 < n < 4$. Since the form is just a single variable n, the first step is essentially nonexistent; it's just a matter of setting n as the thing we're testing and then seeing if it fits the properties. First, we look at $n = 1$. In this case, $n \in \mathbb{Z}$ is true because $1 \in \mathbb{Z}$ and $0 < n < 4$ is true because $0 < 1 < 4$. So $1 \in D$. So far so good. Now, look at $n = 2$. $2 \in \mathbb{Z}$ and $0 < 2 < 4$,, so $2 \in D$ as well. Lastly: $3 \in \mathbb{Z}$ and $0 < 3 < 4$, so $3 \in D$.

(g) $D \subseteq B$

Yes! This is a little tougher to show because we don't have a list of the members of D. So we have to look at the conditions for a given member of D. If something is an member of D then it is an integer, bigger than 0, and smaller than 4. Since we're talking about integers and we have a lower limit and an upper limit, let's just start at the bottom and work our way up, until we hit the top. We don't have to check 0 or smaller integers because none of them are greater than 0, so they aren't in D. The first integer bigger than 0 is 1, and that is also an member of B. The next integer is 2, and $2 \in B$ as well. Next, $3 \in B$. Once we hit 4, though, we've hit the upper limit. $4 \nleq 4$, so it's not an member of D, and so we don't have to test it. Anything bigger than that is not less than 4, so we don't have to check them either, and we're done.

(h) $B = D$

Yes! If you were paying attention to the previous two problems, this should be immediately clear. $B \subseteq D$ means that every member of B is also an member of D. $D \subseteq B$ means that every member of D is also an member of B. So they must have the same members. In other words, they are the same set! In fact, this is one good way to show that two sets are equal (especially if set-builder notation is involved): first show that the first set is a subset of the second, and then show that the second set is a subset of the first.

(i) $B \subsetneq D$

No. Because even though $B \subseteq D$, we know that they are equal.

(j) $A \subseteq \mathbb{N}$

Yes. Each member of A (0, 1, 2, and 3) is a natural number, so they are all members of $\mathbb{N}$.

(k) $A \subseteq S$

No. While some of the members of A are in S, not all of them are. In particular, $2 \in A$, but you can't match $2 = n^2$ unless you allow n to not be an integer, so $2 \notin S$.

(l) $S \subseteq \mathbb{Z}$

Yes. Suppose we had an member of S. Let's give it a name, just to make talking about it easier. We could call it x if we wanted to, but since it's an member of S, we know it can be written $x = n^2$ for some integer n. Since n is an integer, so is $x = n^2$, and therefore $x \in \mathbb{Z}$. This works for any member of S, so we've just shown that every member of S is also an member of $\mathbb{Z}$. In general, this is how you go about proving that one set is a subset of another (especially if they're in set-builder form). Give a name to an member of the subset, just so you can talk about it. Because it's an member of the subset, it must satisfy certain properties. Use those facts to prove that it satisfies the properties for the superset. Since this argument works for *any* member of the subset, you've just proven that it is a subset.

(m) $\mathbb{Z} \subseteq \mathbb{Q}$

Yes. Suppose $n \in \mathbb{Z}$. We need to show that $n \in \{\frac{p}{q} \mid p \in \mathbb{Z} \wedge q \in \mathbb{Z} \wedge q \neq 0\}$ (that's the definition of $\mathbb{Q}$). So we need to write n in the form $\frac{p}{q}$. Yesterclass, we talked about how to write an integer as a fraction. The easiest way is just to take the integer (say 5) and put it over 1 (setting $p = 5$ and $q = 1$, we get $5 = \frac{5}{1}$), and that's exactly what we're going to do here, only we don't know exactly which integer n is. But that's okay. All we need to know is that it is an integer. So we set $p = n$ (whatever n is) and $q = 1$. That's the matching part. Since $n \in \mathbb{Z}$, so is p, and we know already that $q \in \mathbb{Z}$ (because $1 \in \mathbb{Z}$) and $q \neq 0$ (because $1 \neq 0$). So $n \in \mathbb{Q}$.

(n) $S \subseteq \mathbb{Q}$

Yes. Since we showed that every member of S is also an member of $\mathbb{Z}$ and every member of $\mathbb{Z}$ is also an member of $\mathbb{Q}$, it follows that every member of S is also an member of $\mathbb{Q}$. This always works. If $E \subseteq F$ and $F \subseteq G$, then we know that $E \subseteq G$. This is *not* true for the $\in$ relation, only the $\subseteq$ relation.

(o) $B \subseteq X$

No, because (for example) $1 \in B$, but $1 \notin X$. All the members of X are sets, and 1 is not a set. In order for B to be a subset, its members have to be members of X, but they don't even have the same kind of members, so there's no hope here.

(p) $B \in X$

Yes. Remember, that $\in$ implies that B *itself* is an member of X, which is true.

(q) $\{\{1\}, \{2\}\} \subseteq X$

Yes. There are two members of the left set: $\{1\}$ and $\{2\}$. Each of those members is also an member of X, so the left set is a subset of X. Notice that the members of the left set are the same kind of thing (sets in this case) as the members on the right.

(r) $\varnothing \subseteq A$

Yes, but only in the same stupid way that an implication is true if its hypothesis is false. Saying "every member of $\varnothing$ is also an member of A" is just like saying "if something is an member of $\varnothing$, then it is also an member of A." Since nothing is ever an member of $\varnothing$, "something is an member of $\varnothing$" is always false, and so this implication is always true, in a trivial sort of way. Another way to think about it is to notice that checking whether some set is a subset of another set requires one check for every member of the first set. So if that set has 2 members, there are two things to check. If it has 1 member, there's only one thing to check. If it has no members, there is nothing to check, so it's automatically false. You'll never find a counterexample to disprove the claim, so it is true. If you understand this, it's easy to see that $\varnothing$ is a subset of *every* set. Even if you don't understand it, it's not hard to remember.

(s) $A \subseteq \varnothing$

No. There are members of A to check, and obviously, none of them are members of $\varnothing$. To pick one at random: $2 \in A$, but $2 \notin \varnothing$.

(t) $\varnothing \subseteq \varnothing$

Yes. Again, this is kind of trivial.[35] There are no members to check from the left set, so it's automatically true.

(u) $\varnothing \subseteq X$

Yes. I don't even have to look at what X is to test this.

(v) $\varnothing \in X$

Yes, but not for the same reason as before. It's true because $\varnothing$ happens to be one of the sets listed as an member of X.

(w) $\varnothing \in A$

No. The short answer is: "because $\varnothing$ isn't one of the items listed in the definition of A." But it's worse than that. The members of A are all numbers, but $\varnothing$ is not a number. It's a set! It's not even the right *type* of thing to be a member of A. The main reason students get this wrong is because they mistakenly memorize something like "the empty set is in all sets," and then get confused about the fact that this is only true if "in" means "a subset of." It's very much *not* true if "in" means "a member of." If you must memorize something, memorize "the empty set is a *subset* of every set." Or better yet, don't memorize it at all, and learn *why* it's trivially true (it's a universal claim where the premise is impossible).

Suppose X and Y are sets. **If you want to justify that X is a subset of Y, you need to somehow justify that every member of X is also a member of Y.** The claim $X \subseteq Y$ is a universal claim, which means you need to show a lot if you want to prove it. If X is a small finite set, you can probably do this just by going through all the members of X one by one and demonstrating that they are all also members of Y. But if X is an infinite set, or even if X is just a big set, you will need to give some sort of argument that every member of X is also a member of Y. Later, we'll discuss how to *prove* such a claim, but if I just ask for an explanation or a justification, then all you need to do is give a few sentences explaining why you think it's true.

[35] In mathematics, "trivial" doesn't mean easy or unimportant. Essentially, it means "true because of some stupid technicality".

Since the claim $X \subseteq Y$ is universal, that means that the claim that $X \nsubseteq Y$ is existential, and only requires one counterexample to prove it. **If you claim that X is *not* a subset of Y, then your justification should consist of giving an example of a member of X that is not a member of Y.** There might only be one such example. Or there might be lots of them. It might even be true that *none* of the members of X are members of Y, but even if this is the case, all you really need to do to disprove the claim that $X \subseteq Y$ is one single counterexample.

2.5.1 The Power Set

Question. How many members are there in the set $B = \{1, 2, 3\}$? List all of the subsets of B. How many subsets are there?

Answer. There are 3 members of B: 1, 2, and 3.
Here are the subsets: $\varnothing$, $\{1\}$, $\{2\}$, $\{3\}$, $\{1, 2\}$, $\{1, 3\}$, $\{2, 3\}$, $\{1, 2, 3\}$
There are 8 of them, provided you remember the tricky ones $\varnothing$ and B itself.

In any theoretical field, whenever there's a group of objects that someone is interested in, they will usually collect all those objects together as members of a set. This is true even if those objects themselves are also sets. Since we often find it useful to talk about all of the subsets of a set, and so it's common to collect all the subsets of a particular set together into a single set of subsets.

Definition 2.1. If you have a set A, then the set of all subsets of A is called the **power set** of A, and it is written $\mathcal{P}(A)$, with a script or cursive capital "P". In other words, $\mathcal{P}(A) = \{B \mid B \subseteq A\}$.

For example, let's look at the power set of the set $E = \{a, b\}$. Every set that is a *subset* of E is an *member* of $\mathcal{P}(E)$. So because $\{a\} \subseteq E$, we know that $\{a\} \in \mathcal{P}(E)$. E only has 4 subsets, so it's not hard to write out the power set of E in set-list notation: $\mathcal{P}(E) = \big\{\varnothing, \{a\}, \{b\}, \{a, b\}\big\}$.

If you can switch between subsets of a set and members of its power set without getting confused, then you're way ahead of the game. This will take some getting used to for most people.

Example 2.7. Let's do a few examples to get used to this definition. Let $A = \{0, 1, 2, 3\}$, $B = \{1, 2, 3\}$, $F = \{n \mid n < 10 \land n \in \mathbb{N}\}$, and $S = \{n^2 \mid n \in \mathbb{Z}\}$.

(a) Is $1 \in B$?

Yes. (Just a warm-up question.)

(b) Is $1 \in \mathcal{P}(B)$?

No. 1 is not even a set, so it cannot be a subset of B.

(c) Is $\{1, 3\} \in B$?

No. (Just checking to make sure you're paying attention.)

(d) Is $\{1,3\} \in \mathcal{P}(B)$

Yes. $\{1,3\}$ is a *subset* of B, so it is a *member* of $\mathcal{P}(B)$.

(e) Give examples of four different members of $\mathcal{P}(A)$.

$\{0,2,3\}$, $\{2\}$, A, $\varnothing$ (Any subset of A is a valid example.)

(f) Give examples of three different members of $\mathcal{P}(S)$.

$\{4,25,100\}$, $\{9\}$, $\{n^4 \mid n \in \mathbb{Z}\}$ (As with the last problem, any subset of S is a valid example.)

(g) Write $\mathcal{P}(A)$ in set-list notation.

$\mathcal{P}(A) = \big\{\varnothing, \{0\}, \{1\}, \{2\}, \{3\}, \{0,1\}, \{0,2\}, \{0,3\}, \{1,2\}, \{1,2\}, \{2,3\}, \{0,1,2\}, \{0,1,3\}, \{0,2,3\}, \{1,2,3\}, \{0,1,2,3\}\big\}$

(h) What is the cardinality of $\mathcal{P}(A)$?

$|\mathcal{P}(A)| = 16$

(i) Write $\mathcal{P}(F)$ in set-list notation.

Just kidding! F has 10 members, and so there are over a thousand subsets of F. I'm not going to ask you to write out a thousand subsets. But that brings up a question that we can answer...

Question. If we know the cardinality of a set, can we write a formula for the number of subsets it has?

Fact. If A is a finite set with cardinality n, then it has 2^n subsets. In other words, for any finite set A, $|\mathcal{P}(A)| = 2^{|A|}$.

In fact, this is probably the reason why we call it the "power" set; the size of the power set of A is 2 raised to the *power* of the size of A.

This is the same formula as for the number of truth assignments (or truth-table rows) for a propositional logic formula with n variables. And the reason for the formula is the same. With the truth assignments, there were 2 possibilities for each variable: set it to "true" or set it to "false". Adding an extra variable meant doubling the number of assignments because you got all the old assignments plus "new variable = T" *and* all the old assignments plus "new variable = F". The same thing happens with subsets. For each member of the set A, there are 2 possibilities: include it in the subset, or don't include it in the subset. If you add one more member to A, you double the number of subsets because you get all the old subsets (without adding the new member) *and* all the old subsets with the new member included.

With infinite sets, it's pretty easy to see that there are infinitely many subsets of any infinite set. And if all you care about is whether the set is finite or infinite, then that's good enough. But if you're interested in infinite cardinalities...

Fact. For *any* set A (including infinite sets), the cardinality of the power set of A is always strictly larger than the cardinality of A. In other words, $|\mathcal{P}(A)| > |A|$.

I will not include the proof here, but it's very similar to the proof that $|\mathbb{N}| < |\mathbb{R}|$, only you can't draw the cute little diagonalization picture for the cases where A itself is already uncountable.

One consequence of this fact means that any countably infinite set (such as $\mathbb{N}$) has *un*countably many subsets. In fact, it turns out that there are the same number of real numbers as there are subsets of natural numbers, i.e., $|\mathbb{R}| = |\mathcal{P}(\mathbb{N})|$.

This also answers the question of whether there are sets that are even bigger than the real numbers because there are strictly more sets of real numbers (members of $\mathcal{P}(\mathbb{R})$) than there are real numbers (members of $\mathbb{R}$).

As I've mentioned before, you do *not* need to know the difference between the different kinds of infinite cardinalities in the non-honors section of the class.

If all this seems too easy...

The power set is a simple definition, but because it forces us to think of the *subsets* of A as *members* of $\mathcal{P}(A)$, it's very easy to get things mixed up.

Example 2.8. Here are a few more challenging problems, designed to test your ability to keep things straight when stuff gets tricky. Let $B = \{1, 2, 3\}$ and $X = \big\{\varnothing, \{1\}, \{2\}, \{1, 2\}\{1, 2, 3\}\big\}$.

(a) Give an example of a member of $\mathcal{P}(X)$.

$\big\{\{1\}, \{1, 2\}\big\}$ or $\big\{\{1, 2\}\big\}$ or $\varnothing$ or X (Those last three answers are kind of smart-alecky, but they are technically correct.)

(b) Give an example of a proper subset of $\mathcal{P}(B)$ that has at least two members.

$\big\{\varnothing, \{2\}, \{2, 3\}\big\}$ (By asking for a *proper* subset and requiring at least two members, I'm just ruling out some of the more trivial answers. What really makes this tricky is that I'm asking for a *subset* of a set whose members are *already subsets of another set.*)

(c) $\{1, 3\} \subseteq \mathcal{P}(B)$?

No. $1 \in \{1, 3\}$, but $1 \notin \mathcal{P}(B)$. (Remember to give a counterexample whenever you need to justify that one set is not a subset of another. In this case, $\{1, 3\}$ is a set of numbers and $\mathcal{P}(B)$ is a set of sets, so there's no hope for $\{1, 3\}$ to be a subset of $\mathcal{P}(B)$, but you should still point out the counterexample. This is kind of like a type-mismatch error.)

(d) $\big\{\{0, 1\}, \{1\}\big\} \subseteq \mathcal{P}(B)$?

No. $\{0, 1\} \in \big\{\{0, 1\}, \{1\}\big\}$, but $\{0, 1\} \notin \mathcal{P}(B)$. ($\{0, 1\}$ is not a member of $\mathcal{P}(B)$ because $\{0, 1\}$ is not a subset of B. In this case, there's no type mismatch; it's just a simple case of 0 not being a member of B.)

(e) $\varnothing \in B$?

No. (Just making sure you're paying attention.)

(f) $\varnothing \subseteq B$?

Yes, because $\varnothing$ has no members, it is trivially a subset of any set, including B.

(g) $\varnothing \in \mathcal{P}(B)$?

Yes, because $\varnothing \subseteq B$. (See previous question.)

(h) $\varnothing \subseteq \mathcal{P}(B)$?

Yes, because $\varnothing$ has no members, it is trivially a subset of any set, including $\mathcal{P}(B)$.

(i) $B \in B$?

No. (This is not a tricky question unless you try to make it tricky.)

(j) $B \subseteq B$?

Yes, every member of B is clearly a member of B.

(k) $B \in \mathcal{P}(B)$?

Yes, because $B \subseteq B$.

(l) $B \subseteq \mathcal{P}(B)$?

No, because $2 \in B$, but $2 \notin \mathcal{P}(B)$. (Again, this is a type mismatch problem. 2 is not a set, so it can't be a subset of B, and so it can't be a member of the power set of B.)

If you really want a headache, try to think about the subsets of $\mathcal{P}(X)$.

3 First-Order Logic

We've been talking a lot about "universal" and "existential" claims in an informal kind of way. They are an essential component of any kind of logical reasoning, and so it makes sense that we would try to capture that sort of thinking into *formal* logic. The result of adding these ideas to Propositional Logic is called **First-Order Logic**, often abbreviated FOL. Sometimes First-Order Logic is also called **Predicate Logic** or **Quantifier Logic**.

First-Order Logic retains all the connectives from Propositional Logic, so we'll still use $\wedge$, $\vee$, $\rightarrow$, $\neg$,... to build up larger formulas from smaller formulas, but in order to accommodate the universal and existential quantifiers, we can't get away with using variables to stand in for atomic propositions anymore. It's no longer enough to just talk about whether a statement is true or false; we need to be able to talk about *which things* a statement is true for and which things is it false for.

Our basic building blocks will come in two forms: **terms**, which stand in for individual objects or values, and **predicates** which stand in for properties which will be true for some objects and false for others. In Propositional Logic, we might assign a single propositional variable S to the statement "4 is even," but in First-Order Logic, we'll break that down a bit further. We might assign a predicate symbol $E(-)$ to mean "$-$ is even," and then assign a constant term f to the number "4". Then we could translate "4 is even," as the formula $E(f)$.

In the formula $E(f)$, E is a **predicate**[36], and f is a **constant term**. A **constant term** stands in for one specific object or value. In practice, we don't really use constant terms very often. When we're being formal, there isn't much interesting to say about formulas that use constant terms. And when we're being informal, we tend not to assign specific letters to constant values, since we can just write $E(4)$, and everyone will know what we mean.

So I'm not going include *constant* terms when we work with formulas of First-Order Logic. But we are going to need **variable terms** if we want to say anything interesting. In particular, they are useful when we want to express universal or existential claims.

It's traditional to use uppercase letters for predicate symbols and lowercase letters for terms. In particular, we usually use lowercase letters near the end of the alphabet like x, y, and z for *variable* terms.

To express universal or existential claims, we are going to introduce two new symbols, called **quantifiers**. For universal claims, we are going to use the **universal quantifier**, which is written like an upside-down, uppercase "A": $\forall$ (LaTeX: `\forall`, Unicode: `U+2200`, HTML: `&forall;`). For existential claims, we are going to use the **existential quantifier**, which is written like a backwards, uppercase "E": $\exists$ (LaTeX: `\exists`, Unicode: `U+2203`, HTML: `&exist;`). When you are reading a formula out loud, the symbol $\forall$ is read as "for all", and the smbol $\exists$ is read as "there exists a/an".

You create a universal formula by starting with the universal quantifier $\forall$, followed by a variable term (like x). Then you make your universal claim about your variable using predicate symbols and maybe some other connectives. So if $E(-)$ means "$-$ is even," then the formula $\forall x E(x)$ would mean something like "Everything is even." Existential claims work in a similar fashion, so $\exists x E(x)$ means something like "Something is even." Although, to be more precise, $\exists x E(x)$ might be better translated as "There is at least one thing that is even."

In many places where people talk about First-Order Logic, it's common to assume that when someone makes a statement about "some" object existing, they are *not* making any statement about whether there is *more than one* such object. So a sentence like "Someone touched my computer," means the same thing as "At least one person touched my computer," which means the same thing as "There exists a person who touched my computer." We will make this assumption in this class. So you do not need to always specify that you mean "at least one"; just saying "some" or "there is a" is fine.

[36] It's a **unary predicate** if you want to be specific. We'll talk more about the difference between unary and binary predicates later.

3.1 Translations for Simple FOL Formulas

If we're going to do translations between English and First-Order Logic, we need to establish some definitions, just like we did with Propositional Logic translations. The most obvious thing we need to define is the meaning of our predicate symbols. I used blanks in my earlier definition of E, but it's better to use variables because very soon we'll talk about *binary* predicates, which will have more than one "blank" in them. So we might define $E(x)$ as "x is even," and $P(x)$ as "x is positive.

Now our translations are going to come out pretty vague-sounding if we just keep talking about "everything" or "at least one thing", so in addition to establishing definitions for the predicate symbols, it's good practice to establish a general **domain of discourse**. A domain of discourse (or just a domain) is a lot like the *universal set* we talked about in the set-theory section. In fact, we will often use the words "universe" and "domain" interchangeably. When we are doing translations, a **domain** or **universe** is just the group of objects that we are pulling from when we talk about "everything" or "something".

So if I say something like "The universe is the set of all integers," then a formula like $\forall x E(x)$ would be more precisely translated as "Every integer is even." This is an obviously false statement because 17 is an integer that is not even. But it's important to be able to translate both true and false statements.

If we want to be able to determine whether a statement is true or false, then our predicates need to be precisely defined, and our domain of discourse needs to be a specific set of well-defined values. But if all we want to do is translation, then we can be a bit more vague. So we can get away with a domain definition like "The universe is some set of words," without specificying exactly which set of words we mean, and we can make predicate definitions like "$V(x)$: x is a verb," without worrying about whether there are words that can be both nouns and verbs. When we're doing translation problems in this class, we'll probably do plenty of examples where we only are precise enough in our definitions so that we can do the translations.

Example 3.1. So let's set up some definitions and do a few examples of translation from English into First-Order Logic. Let the universe be some set of English words. Define $F(x)$ as "x has four syllables," and $V(x)$ as "x is a verb."

(a) "Every word has four syllables."

$\forall x F(x)$ (Note that the actual variable we use doesn't matter, so the formula $\forall w F(w)$ would also be a correct translation. All that matters is that the variable attached to the quantifier is the same as the variable that appears with the predicate. It's traditional to use x for the first variable, but it is not required.)

Of course, whether this statement is true or false depends on exactly which words are in the universe. And that matches what we would expect from the English sentence; its truth depends on context. If someone was just talking about English in general and said "Every word has four syllables," we'd all know this was false because there are lots of English words that don't have four syllables, like "potato", which only has three syllables. But if someone was looking at a specific list of words that they found written on a piece of paper and they said "Every word has four syllables," then that might or might not be true, depending on what was written on the paper.

(b) "Some word has four syllables."

$\exists x F(x)$

(c) "Some verb has four syllables."

$\exists x\big(V(x) \land F(x)\big)$

For this formula to be satisfied, you'd need to have one specific word that is a verb *and* has four syllables.

Notice that the word "and" doesn't actually appear in the original English sentence. But $\land$ is definitely the correct translation here. This sentence means the same thing as "There is a word that is a verb *and* that has four syllables." This magical "and" appears as a consequence of how the word "some" works in English. Actually, this isn't unique to the word "some" or even to the English language. Pretty much any way you can express the existential quantifier in any natural human language will have an implicit "and" in it (e.g., "There is a verb with four syllables," or "There exists at least one verb that has four syllables.")

When we translate into the formal language of First-Order Logic, we have to make that hidden "and" into an explicit $\land$. You can*not* try to keep it implicit. Sometimes I see students try things like $\exists x\big(V(x)F(x)\big)$ or even $\exists x F\big(V(x)\big)$, but those aren't even formulas. They're just nonsensical strings of symbols that kind of look like formulas.

(d) "There is a verb, and there is a word with four syllables."

$\exists x V(x) \land \exists x F(x)$

Note that there's no reason to assume that the x from the first part and the x from the second part are the same word. This formula is satisfied as long as the universe has at least one verb and at least one four-syllable word. The verb and the four-syllable word might or might not be the same word. If we really wanted to, we could use two different variables for the translation, e.g., $\exists x V(x) \land \exists y F(y)$, and that would also be a perfectly acceptable translation. But it's pretty common to reuse variables like this when we can get away with it.

Those last two examples do not have the same meaning, and they're a good introduction to talking about some important syntactic features of First-Order Logic. Remember that **syntax** is a word that describes the "grammar" of a language, not its meaning. Things like order of operations, where parentheses need to go, which connectives go between formulas (like $\wedge$ and $\vee$), and which go in front of formulas (like $\neg$) are issues of syntax.

In the order of operations for First-Order Logic, a quantifier like $\exists x$ gets a higher priority than the Propositional Logic connectives like $\wedge$. So If you start a formula with $\exists x V(x) \wedge \cdots$, then that first $\exists x$ quantifier only applies to $V(x)$, and everything from the $\wedge$ on is unaffected by that quantifier.

To put it more precisely, we say that the **scope** of the first $\exists x$ in the formula $\exists x V(x) \wedge \exists x F(x)$ is just $V(x)$. And the scope of the second $\exists x$ is just $F(x)$. If we were to draw in optional parentheses to make it clearer, we might rewrite the formula as $\big(\exists x V(x)\big) \wedge \big(\exists x F(x)\big)$. The fact that the first $\exists x$ only applies to $V(x)$ is why we're allowed to reuse the variable x again in the the second part of the formula.

In the formula $\exists x\big(V(x) \wedge F(x)\big)$, we've used explicit parentheses to make it clear that the *scope* of the quantifier $\exists x$ is the entire rest of the formula: $V(x) \wedge F(x)$. If we left off those parentheses, we'd end up with a very confusing and technically incomplete formula: $\exists x V(x) \wedge F(x)$. In this "formula", the scope of the quantifier $\exists x$ is $V(x)$. We say that the quantifier $\exists x$ **binds** the x that occurs in $V(x)$. But the $\wedge F(x)$ is *outside* of the scope of the quantifier $\exists x$. So the x in $F(x)$ is **unbound**, and we say that it is a **free** variable.

A formula that has free variables in it is not a **well-formed** formula. We can't give a translation for it or determine when it is true and when it is false because there's no meaning assigned to x. In this case of this example, the free variable was the result of an error. But free variables are not always bad things. For example, the formula $V(x) \wedge F(x)$ has two free variables in it. It's definitely not a *well-formed* formula, but not because of an error. It's just not a *complete* formula *by itself.* But it's still a formula, and we can do things like apply equivalence laws to it ($V(x) \wedge F(x) \equiv F(x) \wedge V(x)$) or use it as a building block in a larger, well-formed formula, as long as we remember to add a quantifier that binds all of its free variables.

In the previous example, it was pretty easy for a human being to guess that the formula $\exists x V(x) \wedge F(x)$ was *supposed* to mean $\exists x\big(V(x) \wedge F(x)\big)$, but leaving out those parentheses is the sort of thing that might make a computer program spit out a syntax error, so it's important to get those parentheses in there. So make sure you get your parentheses correct when doing translation or you might end up with unbound variables screwing up your answers.

The Special Relationship Between $\forall$ And $\rightarrow$

In an earlier example, we talked about how the existential quantifier often comes with an implicit "and" in it. The difference between "Some word has four syllables," ($\exists x F(x)$) and "Some *verb* has four syllables," ($\exists x\big(V(x) \wedge F(x)\big)$) is a matter of adding an extra conclusion, an additional fact that is true about the four-syllable word. "Some verb has four syllables," means the same thing as "Some word is a verb *and* has four syllables." So $\wedge$ makes sense in this situation.

But the universal quantifier absolutely does not work the same way.

Example 3.2. Translate "Every verb has four syllables," into First-Order

Logic.

The relationship between "Every *word* has four syllables," ($\forall x F(x)$) and "Every *verb* has four syllables," is *not* a matter of adding an extra conclusion. "Every verb has four syllables," absolutely does not mean the same thing as "Every word is a verb *and* has four syllables." It's not even close. "Every word is a verb and has four syllables," (which would be translated as $\forall x\big(V(x) \wedge F(x)\big)$) is a much stronger claim than "Every word has four syllables." The "and" version requires that every word be *both* a verb *and* a four-syllable word. But "Every *verb* has four syllables," is a much *weaker* claim than "Every word has four syllables". The "every verb" version doesn't tell us anything about which words are verbs or not. It doesn't even require *all* the words to have four syllables. It only tells us that *if* a word is already a verb, *then* we can conclude that it has four syllables.

In other words, "Every verb has four syllables," would be translated as $\forall x\big(V(x) \to F(x)\big)$. "Every verb has four syllables," means the same thing as "For every word, if it is a verb, then it has four syllables."

So while an existential quantifier word typically comes with a hidden "and", a *universal* quantifier word (like "every" or "all") usually comes with a hidden "if-then". As before, this happens with lots of different phrasings for the universal quantifier (e.g., "All verbs have four syllables," or "Any word that is a verb must have four syllables,"), and it is not limited to English.

Also as before, we have to make the $\to$ explicit in the formula. NEVER WRITE A FORMULA LIKE $\forall x\big(V(x)F(x)\big)$ OR $\forall x F\big(V(x)\big)$.

General Principle. The universal quantifier goes hand-in-hand with implication. You can't really have one without the other.

The universal quantifier pretty much always comes with an implication hiding inside of it. The only time we can get away with using $\forall$ without $\to$ is if the group of things we are talking about is the entire universe. So we don't need $\to$ to talk about "every *word*" if the universe is all the words. And we don't need $\to$ to translate a sentence about "all numbers" if the universe is all of the numbers. But if you were to talk about "all even numbers," you would absolutely need to use something like $\forall x\big(E(x) \to \cdots\big)$.

In fact, every implication (even the ones we used back when we were just doing Propositional Logic) kind of has a universal quantifier hiding inside of it. When we say something like "if the lever is pushed, the alarm will sound," what we really mean is that *every* time the lever is pushed, the alarm will sound. This is possibly why it sometimes feels awkward to try and determine whether an if-then sentence is true or false when you only get to look at one example.

Now it's entirely possible for $\wedge$ to appear with $\forall$, but it's not going to be hidden. A formula like $\forall x\big(V(x) \wedge F(x)\big)$ needs to have something in it that indicates that there are two separate requirements for every word. So you could translate this as "Every word is a verb and has four syllables," or "All words are verbs that have four syllables." You end up with a very strong statement when you use $\forall$ with $\wedge$ like this, but it does crop up every now and then.

Don't Mix $\rightarrow$ With $\exists$

On the other hand, you really should *not* try to mix $\rightarrow$ with $\exists$. You really don't want to be using $\rightarrow$ without a universal quantifier around to go with it. A formula like $\exists x\big(V(x) \rightarrow F(x)\big)$ is *technically* a well-formed formula, but it doesn't really correspond to the standard meanings we usually give to the words "if" and "then". I mean you could *try* to translate it as "There is a word such that if that word were a verb, it would have four syllables," but what does that sentence even mean?

If you look at the circumstances in which the formula $\exists x\big(V(x) \rightarrow F(x)\big)$ is satisfied, you'll realize that if there's even a single non-verb in the universe, the sentence must be true. For example, if "potato" is in the universe, then the claim $V(\text{potato})$ is false because "potato" is not a verb. Similarly, $F(\text{potato})$ is false because "potato" doesn't have four syllables. So $V(\text{potato}) \rightarrow F(\text{potato})$ evaluates to $\text{F} \rightarrow \text{F}$, which is true. So technically, if there's a single non-verb in the unverse, then $\exists x\big(V(x) \rightarrow F(x)\big)$ is true. Similarly, if there's a single four-letter word in the universe (even if it's not a verb), then $\exists x\big(V(x) \rightarrow F(x)\big)$ is true.

This might not feel right to us because if we try to translate $\exists x\big(V(x){\rightarrow}F(x)\big)$ using "if", we're dragging a universal claim into the formula that isn't supposed to be there. The only way I can think of to get a sensible translation out of a formula like this is to find a way to translate $\rightarrow$ that doesn't sound like implication at all. Fortunately, we can use the equivalence rule of Material Implication $p{\rightarrow}q \equiv \neg\, p{\vee}q$ to help us out here. This tells us that $\exists x\big(V(x){\rightarrow}F(x)\big)$ is logically equivalent to $\exists x\big(\neg\, V(x) \vee F(x)\big)$. So we could translate this formula as "There is a word which either is not a verb or does have four syllables." It's not a *direct* translation, but it is a valid translation.

Now I promise that I won't ask you to translate a formula like $\exists x\big(V(x) \rightarrow F(x)\big)$ on quizzes or tests in this class. That would just be mean. I might put it on a homework assignment *once*, just to force you to confront how awkward the combination of $\rightarrow$ and $\exists$ is. The important thing is that *you* should also never use a formula like $\exists x\big(V(x) \rightarrow F(x)\big)$ as a translation of an English sentence.

3.2 Models for Simple FOL Formulas

We'll come back to translation to tackle more complex formulas (such as those involving negation) in just a bit, but I want to talk about models first. Way back when we first started talking about Propositional Logic, I introduced you to the idea of a *truth assignment.* A truth assignment is a specific kind of **model**, meaning, that it carries enough information needed to determine whether a logical formula is true or false. Truth assignments are the simplest possible model we can come up with for Propositional Logic, so we use them to establish all our other definitions (satisfiable formulas, tautologies, contradictions, contingencies, consistent sets of formulas, equivalent pairs of formulas, valid arguments, etc.)

For First-Order Logic, we need a little bit more than just a truth assignment to determine whether a formula is satisfied or not. It's not enough to say whether a predicate is true or false. Instead we have to say *which things in the universe* the predicate is true for and which things it is false for. And for that to even make sense, we have to first say what the universe is.

So a First-Order Logic **model** has to consist of a set (to be used as the universe) and an **interpretation** that says which members of the universe make which predicate symbols true and which ones make them false.[37]

In the honors section of the class, we are going to do this formally. That means we are going to define an **interpretation function** (usually called $\mathcal{I}$). This function will assign each predicate symbol a subset of the universe. So $\mathcal{I}(P)$ is the set of all things that make the predicate P true. For example, if your universe is $\mathcal{U} = \{1, 2, 3\}$, and you define $\mathcal{I}(P) = \{1, 2\}$ and $\mathcal{I}(Q) = \{2, 3\}$, then you are saying that $P(1)$, $P(2)$, $Q(2)$ and $Q(3)$ are all true, but that $P(3)$ and $Q(1)$ are false.

More formally:

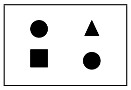
Definition 3.1. A **model** for First-Order Logic is a pair $(\mathcal{U}, \mathcal{I})$ consisting of a universal set $\mathcal{U}$ together with an interpretation function $\mathcal{I}$ that assigns each predicate symbol a subset of $\mathcal{U}$.

In the non-honors class, I am not going to require you to use this kind of notation, and we're not going to use arbitrary sets to interpret our predicate symbols. Instead, we are going to create less abstract, *toy* models that are easier to get our heads around. You wouldn't use models like these if you were doing research into logic itself (such as with automated theorem proving), but they're good enough for learning about First-Order Logic.

For the toy models I'll ask you to create in this class, I will describe the kinds of things we'll use in the universe and give you a *meaningful* interpretation of the predicates as a sort of starting point for your model. Your task will be to fill in the specific objects in the universe to complete the model.

Example 3.3. For the toy models in these examples, the universe will be some set of shapes. Use the definitions $S(x)$: "x is solid," and $C(x)$: "x is a circle."

(a) Give an example of a model that satisfies the formula $\forall x S(x)$, but does not satisfy the formula $\forall x C(x)$.



To make sure the model satisfied $\forall x S(x)$, every shape in the universe had to be solid. To make sure that $\forall x C(x)$ is *not* satisfied, we just had to include *at least one* shape that is not a circle. (I included two.)

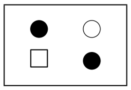
If I wanted to do this as a formal (non-toy) model, I might say something like this:

Let $\mathcal{U} = \{0, 1, 2, 3\}$. Define $\mathcal{I}$ as follows: $\mathcal{I}(S) = \{0, 1, 2, 3\}$ and $\mathcal{I}(C) = \{0, 3\}$. Then the model $(\mathcal{U}, \mathcal{I})$ satisfies $\forall x S(x)$, but not $\forall x C(x)$.

[37] Technically, we'll need a little bit more when we get to binary predicates, but this is enough for unary predicates like we've been using so far.

This is basically the same model as the toy model I drew above, just with all the unnecessary visual details abstracted away. Even the numbers aren't being treated as measuring anything; they're just a convenient source of arbitrary names.

(b) Give an example of a model that satisfies $\forall x\big(S(x) \rightarrow C(x)\big)$ but not $\forall x\big(C(x) \rightarrow S(x)\big)$



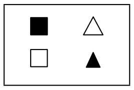
To satisfy $\forall x\big(S(x) \rightarrow C(x)\big)$, every solid shape has to be a circle. To make sure $\forall x\big(C(x) \rightarrow S(x)\big)$ is not satisfied, there has to be at least one circle that is *not* solid.

Here's a formal, set-theoretic version of the same model. The model $(\mathcal{U}, \mathcal{J})$ satisfies the first formula but not the second, where $\mathcal{U} = \{0, 1, 2, 3\}$, $\mathcal{J}(S) = \{0, 3\}$, and $\mathcal{J}(C) = \{0, 1, 3\}$.

In this case, I chose to use the same variable name for the universe $\mathcal{U}$ as in the last example because I used the same values in the universe, but I used a different name $\mathcal{J}$ for the interpretation to emphasize the fact that this is a different interpretation. In general, you don't *have* to change the variable names you use for your universe and/or interpretation function from problem to problem. The only time you really need to use different variable names is if you wanted to directly compare the models. For example, I could say that $(\mathcal{U}, \mathcal{I})$ satisfies $\forall x S(x)$, but $(\mathcal{U}, \mathcal{J})$ does not. Keeping the names distinct could also be useful if you wanted to reuse some of the same models as answers for different problems without rewriting the definitions every time you use them.

(c) Give a model that satisfies $\exists x S(x)$, but not $\exists x C(x)$.

$\mathcal{M}_1$:



To satisfy $\exists x S(x)$, we need at least one solid shape (I included two). To *not* satisfy $\exists x C(x)$, we cannot include any circles.

Formally: Let $\mathcal{M}_1 = (\mathcal{U}, \mathcal{I}_1)$, where $\mathcal{U} = \{0, 1, 2, 3\}$, $\mathcal{I}_1(S) = \{0, 3\}$, and $\mathcal{I}_1(C) = \varnothing$. $\mathcal{M}_1$ satisfies the first formula, but not the second.

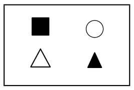
Don't be afraid to set the interpretation of a predicate to the empty set. Some students are so uncomfortable with the empty set that they do all sorts of crazy things, like trying to set $\mathcal{I}_1(C) = \{5\}$, even though 5 is not part of the universe. This is not allowed. The interpretation of C *must* be a subset of the universe. If you don't want anything to satisfy $C(x)$, then don't put anything in the interpretation of C!

In this case, I also gave the model a name: $\mathcal{M}_1$. If you're using the formal notation, this isn't strictly necessary because you can just write something like $(\mathcal{U}, \mathcal{I}_1)$ instead of $\mathcal{M}_1$, but it's common to give single-letter names to models even if you have other ways to refer to them.

You may have noticed that I've been using a calligraphic/script font for the models, universe, and interpretation function. This is not a requirement, but it's a common strategy that helps avoid variable name conflicts. So if you have a predicate called M, you can still have a model named $\mathcal{M}$ without worrying about being confusing, as long as it's easy to tell the difference between how you write a regular M and how you write a script $\mathcal{M}$.

(d) Give a model that satisfies $\exists x S(x) \land \exists x C(x)$, but not $\exists x\big(S(x) \land C(x)\big)$.

$\mathcal{M}_2$:



To meet the first requirement, I needed to make sure that I included at least one solid shape (I included two: a square and a triangle), *and* at least one circle (I included just one white circle). If this was the only requirement, I could have satisfied the formula with a single solid circle, but that would not have worked with the second requirement. To make sure that $\exists x\big(S(x) \land C(x)\big)$ was *not* satisfied, I needed to make sure that there wasn't a shape that was *both* solid *and* a circle.

The formal version: Let $\mathcal{M}_2 = (\mathcal{U}, \mathcal{I}_2)$, where $\mathcal{U} = \{0, 1, 2, 3\}$, $\mathcal{I}_2(S) = \{0, 3\}$, and $\mathcal{I}_2(C) = \{1\}$. $\mathcal{M}_2 \vDash \exists x S(x) \land \exists x C(x)$ but $\mathcal{M}_2 \nvDash \exists x\big(S(x) \land C(x)\big)$.

Remember that the symbol $\vDash$ can be used as a shorthand for the word "satisfies".

(e) Give a model that satisfies $\exists x\big(S(x) \land C(x)\big)$, but not $\exists x S(x) \land \exists x C(x)$.

No such model exists! In order to satisfy $\exists x\big(S(x) \land C(x)\big)$, there has to be a solid circle in the model. And if there's a solid circle in the model, then there is a solid shape in the model (thus satisfying $\exists x S(x)$) and there is a circle in the model (thus satisfying $\exists x C(x)$). So any model that satisfies the first formula must also satisfy the second formula.

In general, when we want to say that every model that satisfies fml_1 also satisfies fml_2, we can write $\text{fml}_1 \vDash \text{fml}_2$. This is the same thing as saying that the following argument is valid:

$$\begin{array}{c} \text{fml}_1 \\ \hline \text{fml}_2 \end{array}$$

In fact, we can carry over all of those semantic definitions from Propositional Logic to First-Order Logic just by replacing the phrase "truth assignment" with the word "model".

I really shouldn't have to repeat all those definitions here, but just for completeness' sake:

Definition 3.2. A formula of First-Order Logic is **satisfiable** iff there is at least one model that satisfies it. It is a **tautology** iff *every* model satisfies it. It is a **contradiction** iff *no* model satisfies it. It is a **contingency** iff there is at least one model that satisfies it and at least one model that does *not* satisfy it.

Definition 3.3. A set of formulas from First-Order Logic is **consistent** iff there is at least one model that satisfies all the formulas in the set.

Definition 3.4. Two pair formulas of First-Order Logic are **logically equivalent** iff every model that satisfies one of the formulas also satisfies the other (and vice versa).

Definition 3.5. An argument of First-Order Logic formulas is **valid** iff every model that satisfies all the premises also satisfies the conclusion.

So for example, because the model $\mathcal{M}_1$ from above satisfies the formula $\exists x S(x)$, we can use $\mathcal{M}_1$ as proof that $\exists x S(x)$ is satisfiable. Similarly, because $\mathcal{M}_1$ does *not* satisfy $\exists x C(x)$, we can use $\mathcal{M}_1$ as proof that $\exists x C(x)$ is *not* a tautology.

And because $\mathcal{M}_2$ satisfies $\exists x S(x) \wedge \exists x C(x)$ but not $\exists x\big(S(x) \wedge C(x)\big)$, we can use $\mathcal{M}_2$ to prove that $\exists x S(x) \wedge \exists x C(x) \not\equiv \exists x\big(S(x) \wedge C(x)\big)$. For the same reason, we can also use the model $\mathcal{M}_2$ to prove that the following argument is *invalid*:

$$\frac{\exists x S(x) \wedge \exists x C(x)}{\exists x\big(S(x) \wedge C(x)\big)}$$

That makes it easy to prove existential claims about First-Order Logic formulas (e.g., a formula is not a tautology, a set of formulas is consistent, an argument is invalid,...), but proving universal claims is much, much harder. Since there are infinitely many possible models (even if you exclude toy models and focus on formal, set-theoretic models), so it's not possible to just build a table to prove that two formulas are equivalent, or that an argument is valid.

We may do a few examples of simple equivalence proofs for First-Order Logic formulas, since we already have most of the tools we need. But we won't be writing any full-fledged proofs to prove that certain FOL formulas are tautologies or to prove the validity of arguments of FOL. You should be prepared to at least give general explanations for certain simple claims, like the one I gave above to explain why every formula that satisfies $\exists x\big(S(x) \wedge C(x)\big)$ must also satisfy $\exists x S(x) \wedge \exists x C(x)$.

3.3 Negation in First-Order Logic

Technically, we don't need to do anything special to allow for negation in our First-Order Logic formulas. It comes in with all the rest of the Propositional Logic connectives. But introducing negation does make FOL formulas harder to *think* about, so it's worth devoting some time to dealing with it. Furthermore, in English, the word "not" has some very complex interactions with words like "every" and "all", which makes translation a headache.

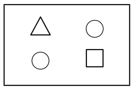
Example 3.4. Let's start with some very simple translation/model exercises. For the translations and for the toy models, we'll set the universe to some set of shapes and use the definitions $S(x)$: "x is solid," and $C(x)$: "x is a circle." As usual, for the formal set-theoretic models, we'll ignore the shapes and just set up arbitrary universes and interpretation functions as needed.

(a) Give a model that satisfies $\exists x \neg C(x)$ and $\forall x \neg S(x)$.

Since the formula $\exists x \neg C(x)$ starts with $\exists$, it is an existential claim. If we're going to satisfy this formula, then we'll need to pick an example of an x that will make $\neg C(x)$ true. So we need to put something in our universe that is *not* a circle. Now we *could* make sure that nothing in our universe is a circle, but that would be working too hard and missing the point. So let's just note that we want to put at least one non-circle in our universe and move on to the second requirement.

The formula $\forall x \neg S(x)$ starts with $\forall$, so it is a universal claim. That means that we've got a requirement that needs to be true for every thing that we put into our universe. To be specific, the requirement that needs to hold for every x is $\neg S(x)$ (i.e., x is not solid). So no matter what thing we want to put in our universe, we have to make sure that the thing is not solid. So let's do that:

$\mathcal{M}_3$:



The formal version: Let $cM_3 = (\mathcal{U}, \mathcal{I}_3)$, where $\mathcal{U} = \{0, 1, 2, 3\}$, $\mathcal{I}_3(C) = \{0, 3\}$, and $\mathcal{I}_3(S) = \varnothing$.

I put the model problem before the translations on purpose. It's entirely possible to break down FOL formulas piece by piece without ever having a global understanding of the entire formula. That might be overkill on simpler formulas like these, where the English translations are relatively simple. But when the formulas get complex enough, natural human languages are just not up to the task. And when they get even more complicated, our limited human brains won't be able to hold on to the entire concept even if we've trained ourselves to think using more precise language. But we can always break the formulas up by analyzing the main connectives and then working our way inward, and then the only limiting factors are our own patience.

But of course, we're nowhere near those limits yet, and for simple formulas like these, it's worth it to try and sum up these requirements into simple English sentences.

(b) Translate $\exists x \neg C(x)$ into English.

We already know that $\exists x C(x)$ can be translated into something like "There is a shape that is a circle." When we introduce the negation, we need to figure out where the "not" is supposed to go in the English sentence. And sometimes you may need to tweak other parts of the sentence to make things match up properly or just to make them sound better. One key to getting your negation into the correct place is to make sure that your end result is still an existential claim. If you try to negate the first "is" in the sentence ("There isn't a shape that is a circle,") you are no longer claiming that something exists, which is why that's not the correct translation. The translation we're looking for is:

"There is a shape that is not a circle," or "Some shape is not a circle," or "There is at least one non-circle."

(c) Translate $\forall x \neg S(x)$ into English.

Following the guideline I mentioned above, we need to make sure that our translation stays a universal claim. So we can definitely rule out something like "Not every shape is solid." That is *not* a universal claim. In fact, "Not every shape is solid," means the same thing as "There is at least one shape that is not solid." So you might be inclined to go with "Every shape is not solid," or "All shapes are not solid." Those should be correct, right?

Eh... kind of? Unfortunately, we've run into a major limitation of English, and it'll be worth it to spend a little time thinking about it. More on this in a bit.

Another approach to the translation is to look back at the work we did with breaking the formula down in the toy model problem, and just building a completely new sentence out of the requirement we came up with there. That might lead you to a translation such as:

"No shape is solid," or "There are no solid shapes," or "There aren't any solid shapes."

And these are good, clear, unambiguous translations of $\forall x \neg S(x)$.

3.3.1 Ambiguities and the Scope of Negation in English

Some of you might be wondering why I have a problem with "Every shape is not solid." Perhaps you think this means the same thing as "No shape is solid." And some of you might be thinking that "Every shape is not solid," means the same thing as "Not every shape is solid," (i.e., "there's at least one shape that isn't solid.")

And that's the problem. The sentence "Every shape is not solid," is **ambiguous**, meaning that its meaning can shift, depending on the context. It can be very hard to see that a sentence is ambiguous when you are the one who wrote it. The writer has a very particular context and intended meaning in their head, but the reader may or may not share that context. I suspect that most people who read these notes in the order I wrote them will be thinking that "All shapes are not solid," and "No shapes are solid," have the same meaning. We were already thinking about the correct interpretation, so it's natural for us to pick this interpretation of the sentence. This is the context represented by the formula $\forall x \neg S(x)$. Here we say that the word "not" has a **narrow scope**, where we are only negating the part about being a solid shape.

The other interpretation might be harder to sympathize with right now. This is the interpretation where "All shapes are not solid," means the same thing as "Not all shapes are solid," and "There's at least one shape that isn't solid." This interpretation is represented by the formula $\neg \forall x S(x)$, simply a denial of the claim that "All shapes are solid." Here we say that the word "not" has a **wide scope**, where we are negating the entire claim that "All shapes are solid."

Most of you probably started with the narrow scope interpretation in mind because context makes that seem more sensible, but I bet I can get you to switch to the wide scope by changing contexts. (Or at least, I hope I can get you to admit that at least *some* people *sometimes* use the wide scope.)

Consider the popular saying "All that glitters is not gold." The narrow-scope reading of this sentence $\forall x\big(\text{Glitters}(x) \rightarrow \neg \text{Gold}(x)\big)$ would mean that *nothing* that glitters is gold, which is clearly wrong. The saying only makes sense if you recognize that gold is glittery. In this case, the wide-scope reading $\neg \forall x\big(\text{Glitters}(x) \rightarrow \text{Gold}(x)\big)$ is the correct one. The saying implies that not every glittery thing is gold, that there are at least some glittery things that are not gold.

So English is messy. How do we deal with that problem? In informal, everyday English, it's usually not a problem. There's usually enough context for people to figure out which meaning is intended. But when context isn't enough, or when you need to be precise (such as when you are translating FOL formulas into English for your discrete structures homework assignment), it's important to avoid ambiguous sentences. In particular, I promise never to ask you to translate a sentence like "Every ____ is not..." or "All ____ do not..." And I'm going to expect the same thing from you. If you translate a formula using a sentence in those formats, it will get marked as "ambiguous", and you won't get full credit for that translation.

Here's how I recommend you approach this specific English ambiguity: any time you find yourself writing "Every ____ is not...", immediately paraphrase that to "No ____ is..." or "Not all ____ are..." to clarify your translation.

3.3.2 The Word "Any"

If I were to poll the class right now about how best to translate the word "any" into First-Order Logic, I think most students would suggest using $\forall$. This is influenced by sentences like "For any non-empty set, there at least two subsets: the empty set and the set itself," or "Any non-empty list can be divided into a first element and the rest of the list." And those *are* universal claims. But the fact that they're universal claims isn't because of the word "any". If you replace "any" with "a", the meanings are still the same: "For a non-empty set, there at least two subsets: the empty set and the set itself," or "A non-empty list can be divided into a first element and the rest of the list." The first example is a universal claim mostly because of the word "for", which acts a bit like "if" in this situation. The fact that the second one is a universal claim has more to do with context than anything else.[38]

In fact, in most places where you see "any", it behaves a lot more like $\exists$ than it does like $\forall$. Certainly, if a sentence starts out "For any…", it's probably going to be a universal claim. But one of the most common uses of "any" is inside the scope of a negation word like "not" (e.g., "There aren't any even prime numbers larger than 2."). In those cases, "any" acts like "some". In fact, if you try to use "some" in a negative context, it will often sound awkward (e,g., "There isn't some even prime number larger than two.") It can also help clear up ambiguities because "any" often doesn't make sense outside of a negative context.[39] So a sentence like "My friend doesn't like someone in their dorm," can be made less confusing by rephrasing it as "My friend doesn't like anyone in their dorm."

Example 3.5. Translate the following sentences into FOL formulas. The universe is some set of numbers. $N(x)$ means "x is negative." $O(x)$ means "x is odd."

(a) "There aren't any odd, negative numbers."

$\neg \exists\big(O(x) \wedge N(x)\big)$

You could also give the translation $\forall x\big(O(x) \rightarrow \neg N(x)\big)$, but that would be a less direct translation. It's correct, but it's not the first translation I'd jump to.

(b) "If any number is negative, then all the numbers are negative."

$\exists x N(x) \rightarrow \forall x N(x)$

3.3.3 An Upgrade for De Morgan's Laws

Example 3.6. Let's do a few more translations and toy models, using the same set-up as before.

(a) $\neg \exists x\big(S(x) \wedge C(x)\big)$

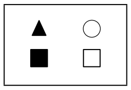
English translations: "There does not exist a solid shape that is a circle," or "There are no solid circles," or "It's not true that there is at least one

[38] If we just start talking about something using "a" or "any" in the context of a CS theory class (e.g., "An NP-complete problem has a Turing Machine that…"), we probably are implying a universal interpretation ("Every NP-complete problem has…"). But if we do the same thing in the context of a narrative or telling a story (e.g., "A man walks into a bar…"), then we're probably implying an existential claim). But even then, there are exceptions (check out the opening line of *Pride and Prejudice* for a notable one).

[39] Technically speaking, the linguistic term **negative context** refers to more than just contexts that are inside of the scope of a negation. For example, the premise of a conditional (e.g., "If any NP-complete problem can be solved by a polynomial time algorithm,…") is also a negative context. One of the most importan jobs of "any" is to act like "some", but it's usually only allowed in negative contexts, so it's very useful for making clear exactly which parts of a sentence are in the negative context and which aren't.

solid circle," or "There aren't any solid shapes that are circles," or "No solid shapes are circles."

Here's a model that satisfies the formula $\mathcal{M}_4$:



Any model that includes a solid circle will *not* satisfy the formula.

(b) $\forall x\big(S(x) \to \neg\, C(x)\big)$

When I encounter a formula that starts $\forall x\big(S(x) \to \cdots\big)$, my first instinct is to start my translation as "Every solid shape…" Finishing out the rest of the formula, I would end up with "Every solid shape is not a circle." And that is correct, or at least it's correct if I use the narrow-scope reading. To clear up the potential ambiguity, I would paraphrase like this:

"No solid shape is a circle."

Look familiar? It should! This formula is logically equivalent to the previous one.

We've seen many times before that a negated universal claim is equivalent to an existential claim (which is why we disprove universal claims by giving counterexamples) and that a negated existential claim is equivalent to a universal claim (which is why we have to work harder to disprove existential claims). We're now in a position to express this relationship more precisely, using the language of First-Order Logic.

Equivalence	Name
$\neg\,\forall x\, p(x) \equiv \exists x\, \neg\, p(x)$	universal negation (De Morgan)
$\neg\,\exists x\, p(x) \equiv \forall x\, \neg\, p(x)$	existential negation (De Morgan)

In essence, these rules are the same as De Morgan's Laws for $\wedge$ and $\vee$. You can basically think of a universal formula as a big fat $\wedge$, with one conjunct for each object in the universe. $\forall x P(x)$ is true if $P(x)$ is true for the first thing in the universe, *and* for the second thing, *and* for the third thing, etc. Similarly, an existential claim is one large $\vee$ formula, with one disjunct for each object in the universe. $\exists x P(x)$ is true if $P(x)$ is true either for the first thing *or* the second *or* the third…

So it shouldn't be surprising that when you negate a universal claim, you get an existential claim. A formula like $\neg\,\forall x P(x)$ says that there isn't a rule requiring $P(x)$ to be true. And that's the same thing as saying that there exists an exception to the rule, i.e., $\exists x\, \neg\, P(x)$ is true.

Similarly, when you negate an existential claim, you get a universal claim. Saying that $\neg\,\exists x P(x)$ is true is the same thing as denying the existence of a thing that makes $P(x)$ true. And if no such object exists, that means that there's a rule that states that no matter which thing you pick, it will not satisfy $P(x)$, i.e., $\forall x\, \neg\, P(x)$.

But the example we were talking about earlier ($\neg \exists x\big(S(x) \wedge C(x)\big) \equiv \forall x\big(S(x) \to \neg C(x)\big)$)is actually a bit more complex. In English, it might seem perfectly obvious, but how did we go from $\wedge$ to $\to$ in the formulas?

Unlike with Propositional Logic, we can't fall back on a truth table to prove that these two formulas are equivalent, but we *can* still use equivalence proofs to get at all the details.

Claim. $\neg \exists x\big(S(x) \wedge C(x)\big) \equiv \forall x\big(S(x) \to \neg C(x)\big)$

Proof.

$$\begin{aligned} \neg \exists x\big(S(x) \wedge C(x)\big) &\equiv \forall x \neg\big(S(x) \wedge C(x)\big) && \text{(Univ. Neg.)} \\ &\equiv \forall x\big(\neg S(x) \vee \neg C(x)\big) && \text{(De Mor.)} \\ &\equiv \forall x\big(S(x) \to \neg C(x)\big) && \text{(Impl.)} \end{aligned}$$

□

And we can write a similar proof to show that $\neg \forall x\big(S(x) \to C(x)\big)$ ("Not every solid shape is a circle,") is equivalent to $\exists x\big(S(x) \wedge \neg C(x)\big)$ ("There is a solid shape which is not a circle.")

Claim. $\neg \forall x\big(S(x) \to C(x)\big) \equiv \exists x\big(S(x) \wedge \neg C(x)\big)$

Proof.

$$\begin{aligned} \neg \forall x\big(S(x) \to C(x)\big) &\equiv \exists x \neg\big(S(x) \to C(x)\big) && \text{(Exist. Neg.)} \\ &\equiv \exists x \neg\big(\neg S(x) \vee C(x)\big) && \text{(Impl.)} \\ &\equiv \exists x\big(\neg\neg S(x) \wedge \neg C(x)\big) && \text{(De Mor.)} \\ &\equiv \exists x\big(S(x) \wedge \neg C(x)\big) && \text{(Dbl. Neg.)} \end{aligned}$$

□

Even though these two equivalences technically required several steps to prove using the more basic equivalences, it's worth becoming very familiar with them on their own right. If you're having a hard time thinking about a formula in it's current form, you can often make it easier to think about by rewriting it in an equivalent form that is easier to understand (or translate).

The most common standard version of a formula is the one where the negation has been "pushed" all the way "inside", so that the only things that are negated are individual predicates, like $\cdots \neg P(x)$ instead of larger, quantified formulas ($\neg \forall x \cdots$). I prefer this form when I'm trying to find models to satisfy a formula, or if I'm trying to see whether two formulas are equivalent or not. It's especialy useful if the formula is so complicated that I need to break it down piece-by-piece to get a better understanding of what it means.

But when I'm doing translations into English, or if I need to find a model that does *not* satisfy a formula, I actually think it's easier to work with the negation "pulled" all the way to the "outside" of the formula, so that the whole formula is just a negation of another formula.

Example 3.7. For the translations and toy models in these examples, let the universe be some set of integers. Use the following definitions. $C(x)$: "x is a perfect cube." $N(x)$: "x is negative." $O(x)$: "x is odd."

(a) Create a model that satisfies the formula $\neg \forall x \Bigl(C(x) \to \bigl(N(x) \land O(x) \bigr) \Bigr)$.

As written, this might be a little hard to think about, but if I do a little bit of manipulation, I can write it in a more convenient form. (Note that this isn't a semi-formal proof, just some scratch work for my own benefit, so I am free to leave out rule names and skip steps.)

$$\begin{aligned} \neg \forall x \Bigl(C(x) \to \bigl(N(x) \land O(x) \bigr) \Bigr) &\equiv \exists x \neg \Bigl(C(x) \to \bigl(N(x) \land O(x) \bigr) \Bigr) \\ &\equiv \exists x \Bigl(C(x) \land \neg \bigl(N(x) \land O(x) \bigr) \Bigr) \\ &\equiv \exists x \Bigl(C(x) \land \bigl(\neg N(x) \lor \neg O(x) \bigr) \Bigr) \end{aligned}$$

In this form, it's much easier for me to break down the requirements for my model. This is an existential claim, so I just have to put one special number into my universe to satisfy the formula. The requirement on my one thing is $C(x) \land \bigl(\neg N(x) \lor \neg O(x) \bigr)$. That's really two requirements: $C(x)$ and $\neg N(x) \lor \neg O(x)$. The special number will have to be a perfect cube to meet the first requirement. And for the second requirement, I can either make it non-negative (i.e., positive or 0), or I can make it non-odd (i.e., even). Let's go with the second option and put the even perfect cube 216 into the universe. I'll add some other numbers just for flavor, but none of them are required to satisfy my formula.

216	-27
10	257

(b) Translate $\neg \forall x \Bigl(C(x) \to \bigl(N(x) \land O(x) \bigr) \Bigr)$ into English.

In general, I think the version that has the negation on the outside is probably easiest for translation. My basic strategy is to just translate the formula without the negation first, and then just negate the whole sentence in English. Negating a whole sentence is quite easy in English. If the non-negated sentence starts with "Every…", then the negated sentence will start "Not every…" If the non-negated sentence starts "There exists a…" or "There is a…", then the negated sentene will start "There does not exist a…" or "There is no…"

In this case, the non-negated formula is $\forall x\Big(C(x) \to \big(N(x) \land O(x)\big)\Big)$. This starts with $\forall x\Big(C(x) \to \cdots\Big)$, so my translation is going to start "Every perfect cube…" What needs to be true about every perfect cube? It needs to satisfy $N(x) \land O(x)$. So the non-negated translation is "Every perfect cube is both negative and odd." Negate the whole thing to get the full translation:

"Not every perfect cube is both negative and odd."

There are other valid translations, including "Not all perfect cubes are odd, negative numbers," or "There is a perfect cube that isn't an odd, negative number," or "There exists a perfect cube that is either not negative or not odd."

In that last example, ambiguity wasn't a big issue, even if you tried to do the translation with the negation pushed all the way in, but in many other cases, pulling the negation all the way out can help you completely sidestep those ambiguity problems I mentioned earlier. For example, a direct translation of $\neg\,\exists x\big(N(x) \land O(x)\big)$ ("There is no negative number that is odd,") doesn't have any issues with ambiguity, while a direct translation of the equivalent formula $\forall x\big(N(x) \to \neg\,O(x)\big)$ ("Every negative number is not odd,") is ambiguous, and would require paraphrasing to make sure it isn't misunderstood.

3.4 Binary Predicates

Adding quantifiers to our concept of logic definitely increases expressiveness, but we're still missing one important piece. We can use (unary) predicates to talk about which things have which properties, but we don't yet have a way to talk about how different things are related to *each other*. This is where **binary** predicates enter the picture.

If you can think of a **unary predicate** as a sentence with a blank in it (e.g., $P(x)$ means "x is prime,") then you can think of a **binary predicate** as a sentence with two blanks in it (e.g., $G(x, y)$ means "x is greater than y.") Where a unary predicate represents a property that some objects have and others don't, a *binary* predicate represents a *relation* that some pairs of objects have, and other pairs don't.

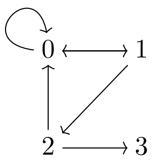
You can also have **ternary predicates** with three blanks, **quaternary predicates** with four blanks, and **n-ary predicates** with n blanks. Full-fledged First-Order Logic allows for predicates with any "arity" you can imagine, but fortunately, there's not a whole lot of interesting things that require looking at predicates with more than two arguments. So in this class, we'll be limiting ourselves to unary and binary predicates. This allows us to express statements like "For every real number, there is a real number that is its inverse," or "There is a Turing machine that can simulate every Turing machine."

3.4.1 Models for FOL with Binary Predicates

We'll talk about translation soon enough, but it'll be nice to have good visual and set-theoretic representations of the relations that can be expressed with binary predicates, so I'd like to talk about those first.

For our toy models, we'll continue to use a variety of objects with different kinds of properties for our unary predicates. You may have noticed that I'm careful to use properties that are independent of each other to ensure that our definitions don't accidentally rule out certain kinds of models. So for example, I might use $P(x)$ to mean "x is positive," and $Q(x)$ to mean "x is even." But I would never use $P(x)$ for "x is positive," and $Q(x)$ for "x is negative," because I'd never be able to make models with objects that are both positive and negative.

If we want our toy models to be as versatile as formal, set-theoretic models, we'll need to be careful about the kinds of relations we use for our binary predicates. If I define $R(x, y)$ as "x is greater than y," then I'm already forcing certain properties onto my universe (for example, if I decide that $R(a, b)$ needs to be true, then I'm not allowed to have $R(b, a)$ also be true). So in order to keep all our options, we're going to keep using the same toy model representation for all my binary relations. When we create our toy models, we will draw arrows between some of the objects in the universe and not between others. We can use $R(x, y)$ to mean that there is an arrow starting at x and pointing to y (i.e., "x points to y.") So our models will look something like this:



In this toy model, we can say that $R(0, 1)$, $R(0, 0)$, $R(1, 0)$, and $R(2, 3)$ are all true, but $R(1, 1)$, $R(3, 2)$, and $R(1, 3)$ are all false.

If we want to represent this toy model as a formal, set-theoretic model, we need a way to represent this idea that some pairs get true and others get false in a set. The way we do this is to interpret a binary predicate not as a set of objects from the universe, but a set of *ordered pairs* of objects from the universe.

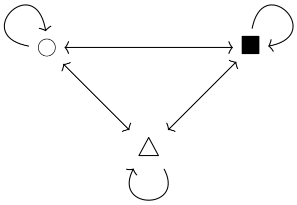
A set of ordered pairs of objects from a set A is called a **relation** on A. We'll talk more about relations in a later section, but for now, just know that they're a useful way to define which pairs are related by a particular predicate and which are not. So the model we defined pictorially above can also be defined set-theoretically as follows.

Let $\mathcal{M} = (\mathcal{U}, \mathcal{I})$, where $\mathcal{U} = \{0, 1, 2, 3\}$ and $\mathcal{I}(R) = \{(0, 0), (0, 1), (1, 0), (1, 2), (2, 0), (2, 3)\}$.

Example 3.8. For the translations and toy models in these examples, let the universe be some set of shapes. Use the following definitions. $C(x)$: "x is a circle." $S(x)$: "x is solid." $P(x, y)$: "x points to y."

(a) Give a model that satisfies $\forall x \forall y P(x, y)$.

A double universal claim like this is pretty easy to get our heads around. For any two objects, there must be an arrow between them. Basically, the only decision we have to make is how many things to include in the universe. Let's go with three.



Formally, this model might look like this: $\mathcal{M}_1 = (\mathcal{U}, \mathcal{I}_1)$ where $\mathcal{U} = \{0, 1, 2\}$ and $\mathcal{I}_1(P) = \{(0,0), (0,1), (0,2), (1,0), (1,1), (1,2), (2,0), (2,1), (2,2)\}$.

The only tricky thing at all here is to keep in mind that the x and y in $\forall x \forall y P(x, y)$ do not necessarily refer to *different* values. So to satisfy this formula, not only does every object have to point to every other object, but they also have to all point to themselves.

(b) Translate $\forall x \forall y P(x, y)$ into an English sentence.

You might be inclined to go with something like "Every shape points at all other shapes," but technically this is not correct. There is nothing in the formula that says the two shapes have to be different for the rule to apply to them. So "Every shape points to all shapes," would be a more correct translation. The only way to translate "Every shape points at all other shapes," into First-Order Logic would be to include = as a symbol in FOL: $\forall x \forall y \big(\neg(x = y) \rightarrow P(x, y)\big)$. In this class, we won't be using = as part of our formal First-Order Logic formulas, so I won't be asking you to translate sentences that talk about "all other" things. And since I won't ask you to translate any formulas with = in them, you shouldn't be using "other" in your English translations either! Not even if it makes it "sound better".

(c) Translate "Every circle points to all of the shapes," into First-Order Logic.

We've already talked about how to translate a sentence that starts off with "Every circle..." already, and the guidelines I gave you earlier still hold here. This is a statement about all of the circles, so, our formula is going to start off with $\forall x \big(C(x) \rightarrow \cdots\big)$. Now what is it that's true for every circle? It has to point to all the shapes. If you're just reading the English sentence from left to right, you might be inclined to get P involved right away, but if we continue with $\forall x \big(C(x) \rightarrow P(x, \cdots)\big)$, we immediately run into trouble. We can't use $P(x, y)$ until both x and y are defined (i.e., until they are both bound by quantifiers). So we need to establish what y is before we can use P. The final translation ends up as:

$\forall x\big(C(x) \to \forall y P(x,y)\big)$

Now it's true that this formula is equivalent to $\forall x \forall y\big(C(x) \to P(x,y)\big)$, so that is also an acceptable translation, but I think it's important to recognize that this is a less direct translation. If I were to translate this formula back into a more literal English sentence, it might come out something like "For any two shapes, if the first one is a circle, then it points to the second shape." That sentence feels a bit odd to me because it's strange to mention the second shape before qualifying that the rule only applies when the first shape is a circle. (Also, someone who reads that sentence might incorrectly assume that the rule doesn't apply when both shapes are the same object. That's wrong because in this sentence the circles need to point to themselves, not just all the other shapes.)

This last example brings up an important point about formulas that have multiple nested quantifiers in them along with unary predicates. If the predicate only mentions the variable bound by the outermost quantifier, then that predicate can be written either inside or outside of the scope of the inner quantifier. Or at least it can in most situations that are likely to arise. So the following equivalences all hold:

$$\begin{aligned}
\forall x\big(P(x) \to \exists y R(x,y)\big) &\equiv \forall x \exists y\big(P(x) \to R(x,y)\big)\\
\exists x\big(P(x) \land \forall y R(x,y)\big) &\equiv \exists x \forall y\big(P(x) \land R(x,y)\big)\\
\forall x\Big(P(x) \to \forall y\big(Q(x) \to R(x,y)\big)\Big) &\equiv \forall x \forall y\Big(P(x) \to \big(Q(x) \to R(x,y)\big)\Big)
\end{aligned}$$

To be more precise, we can express these as equivalence laws:

Equivalence	Name
$\forall y\big(p(x) \to r(x,y)\big) \equiv p(x) \to \forall y\, r(x,y)$	"Quantifier Movement"
$\exists y\big(p(x) \to r(x,y)\big) \equiv p(x) \to \exists y\, r(x,y)$	
$\forall y\big(p(x) \land r(x,y)\big) \equiv p(x) \land \forall y\, r(x,y)$	
$\exists y\big(p(x) \land r(x,y)\big) \equiv p(x) \land \exists y\, r(x,y)$	

Really, the rules don't need to have an x in them at all, but I think that for students at your level, the formulas are easier to read in this format.

I've put the names of these rules in scare quotes because the rules themselves usually aren't given names. This name is one I made up that sounds about right.

Question. Which version is more useful: the one with all the quantifiers out front (e.g., $\exists x \forall y\big(P(x) \land R(x,y)\big)$) or the one where the quantifiers have each been pushed in as far as they can legally go (e.g., $\exists y\big(p(x) \to r(x,y)\big)$)?

Answer. As usual, it depends on what you're doing with the formulas. I personally think that pushing the quantifiers inwards (where you can) makes both translations and model-building much, much easier. But the standard form is to have all the quantifiers all the way out front, so that's probably the best format if you're trying to determine whether two formulas are equivalent or not.

Example 3.9. Using the same definitions as the previous example, translate $\exists x \exists y \big(S(x) \wedge C(y) \wedge P(x,y)\big)$ into an English sentence.

You *could* attempt to translate this directly into English, but I would recommend using the equivalence rules we just talked about to rewrite this formula as $\exists x \Big(S(x) \wedge \exists y \big(C(y) \wedge P(x,y)\big)\Big)$. In this format, I can easily separate out the part of the formula that is only about x: $\exists x \Big(S(x) \wedge \cdots\Big)$. From that part, I can see that the sentence is going to begin with "There is a solid shape that..." or "Some solid shape..." I can also separate out the part that's just about y: $\exists y \big(C(y) \wedge\big)$. And from that, I can see that when I do get to mentioning the second shape, I can say something like "...some circle..." or "...has at least one circle that..." Putting everything together, I can come up with several valid translations of the sentence:

"There is a solid shape that points to at least one circle," or "Some solid shape has a circle that it points to."

I'd probably give full credit to the translation "Some solid shape points to some circle," although that sounds a bit awkward to me.

Note that these rules do *not* imply that you can just throw the quantifiers in wherever you want. Sometimes you can't push the quantifiers in at all. Don't get too eager and try to rewrite $\forall x \exists y \big(C(y) \wedge P(x,y)\big)$ as $\forall x \big(C(y) \wedge \exists y P(x,y)\big)$. That second formula has an unbound y in it, so it's not even a complete FOL formula!

And even when the resulting formulas do make sense, the equivalence rule only applies in those specific instances where you have $\wedge$ or $\rightarrow$, with nothing else getting between. For example, $\forall x \forall y \big(P(x) \rightarrow \neg R(x,y)\big)$ is *not* equivalent to $\forall x \big(P(x) \rightarrow \neg \forall y R(x,y)\big)$. That negation makes a huge difference.

It also makes a difference what *order* the quantifiers go in.

Example 3.10. Is $\forall x \exists y P(x,y)$ equivalent to $\exists y \forall x P(x,y)$?

Absolutely not! Even though in both formulas it's the "every shape" that is doing the pointing and the "some shape" that is being pointed to, the two formulas describe fundamentally different problems. We can prove this by giving a model that satisfies the first formula, but not the second formula.

To make things easier to think about, let's set our universe to be $Univ = \{0,1,2\}$. For the toy model we're going to construct, we'll use the usual interpretation of $P(x,y)$ as "x points to y." For the formal version, we'll define a formal interpretation function.

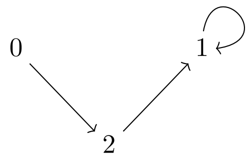
The first formula $\forall x \exists y P(x,y)$ expresses a universal claim. It says that no matter what x you pick from your universe, the formula $\exists y P(x,y)$ is true. In our universe of three elements, that means we have three separate requirements: $\exists y P(0,y)$, $\exists y P(1,y)$, and $\exists y P(2,y)$. All three need to be true to satisfy $\forall x \exists y P(x,y)$.

So how do we satisfy $\exists y P(0,y)$? Well that is an existential claim, so we just need to ensure that there exists at least one value y so that $P(0,y)$ is true. It doesn't really matter much *which* value we pick for y, so let's pick 2. We'll make sure that there is an arrow pointing from 0 to 2. That will make $P(0,2)$ true, and the example of 2 proves that the existential claim $\exists y P(0,y)$ is true.

Next, we need to make sure our model also satisfies $\exists y P(1,y)$. So we need to pick a y and make sure that $P(1,y)$ is true for that specific y. There's absolutely no reason why we need to pick the same y as we did for the last requirement because the three requirements are *separate* requirements. So let's pick 1 as our y this time. (There's nothing that says x and y have to be different!) So as long as we ensure that 1 points to itself, then we are satisfying $P(1,1)$, and if we satisfy $P(1,1)$, then we have satisfied $\exists y P(1,y)$.

Lastly, we need to satisfy $\exists y P(2,y)$. You probably get the picture by now, so I'll just skip ahead and pick 1 for y this time. (We didn't *have* to pick the same y as the last one, but we certainly *can* pick the same one.) So by making sure that there's an arrow from 2 to 1, we've satisfied $P(2,1)$, and in turn, that means we've satisfied $\exists y P(2,y)$.

So our model now looks like this:



The model defined by this picture satisfies $\exists y P(x,y)$ for all three x's in our universe, so this model satisfies the formula $\forall x \exists y P(x,y)$. Let's call this model $\mathcal{M}_2$.

Formally: $\mathcal{M}_2 = (\mathcal{U}, \mathcal{I}_2)$, where $\mathcal{U} = \{0,1,2\}$ and $\mathcal{I}_2(P) = \{(0,2),(1,1),(2,1)\}$.

Since there are no negative universal requirements here, I could add lots of other arrows as well, and it would still satisfy the model, but this is enough for our task because I claim that $\mathcal{M}_2$ does *not* satisfy $\exists y \forall x P(x,y)$.

Why not? Well, let's break down this other formula. The formula $\exists y \forall x P(x,y)$ is an *existential* claim because it starts out $\exists y \cdots$. To make this formula true, all we need to do is find an example of a y that satisfies $\forall x P(x,y)$. So if it turns out that $\forall x P(x,0)$ is true, then $\exists y \forall x P(x,y)$ would be true. Or alternatively, if we've somehow satisfied $\forall x P(x,1)$ or $\forall x P(x,2)$, then we have also satisfied $\exists y \forall x P(x,y)$.

So *have* we satisfied any of these three possibilities? Let's start with $\forall x P(x,0)$: is this true in the model $\mathcal{M}_2$? Well in order for this to be true, we'd need $P(x,0)$ to be true for all three possible x's in the universe. We'd need $P(0,0)$, $P(1,0)$, *and* $P(2,0)$. And we haven't satisfied any of those! So definitely not.

But that's okay, we only needed one of the three formulas to be true in order to satisfy $\exists y \forall x P(x,y)$. Let's look at the second possibility: $\forall x P(x,1)$. For this to be true, we'd need to have $P(0,1)$, $P(1,1)$, and $P(2,1)$ all satisfied. And we almost do! We've got arrows from 1 to itself and from 2 to 1, but there is no arrow from 0 to 1, so $P(0,1)$ is false, and hence $\forall x P(x,1)$ is false.

Similarly, we don't have an arrow from 1 to 2, so the model doesn't satisfy $P(1,2)$ and hence it doesn't satisfy $\forall x P(x,2)$ either.

None of the three possibilities for y result in $\forall x P(x, y)$ being true, so the formula $\exists y \forall x P(x, y)$ cannot be true. So our model $\mathcal{M}_2$ works! It satisfies $\forall x \exists y P(x, y)$, but not $\exists y \forall x P(x, y)$, and therefore the two formulas are not equivalent.

Question. That seemed like a lot of work! Do I need to do all that work every time I want to build a model that satisfies a formula or check if a model satisfies a formula?

Answer. No. All you needed to do for the problem was to *give* the model. The rest was just a demonstration that you always have the *option* to break the formula down piece by piece. As you do more and more of these problems, your intuitions about FOL formulas will get better and better. But if you ever find yourself doubting your intuitions, you can always resort to breaking the formulas up piece by piece as I just did.

In practice, you are unlikely to need to do this kind of breakdown for simple formulas like these. When there are only two quantifiers, there aren't many possibilities, and so you'll get used to them pretty quickly. If both quantifiers are universal or both are existential, there's not much tricky going on at all. So the two interesting cases are when you have a universal quantifier outside of an existential quantifier and vice versa.

3.4.2 Universal-Existential Claims

Any formula that starts out $\forall x \exists y \cdots$ is first and foremost a universal claim. If we want to get more specific about things, it's a **universal-existential claim**. I like to think of these kinds of claims as everyone-gets-a-buddy formulas. To create a model satisfying a formula like this, you need to ensure that for each object in the universe, you assign them a second object. So everyone gets a buddy. You *can* assign everyone a different buddy, or you can assign the same buddy to everyone, or you can assign some people the same buddy and other people different buddies. As long as every person has a buddy, the formula is satisfied.

What does it mean to be a buddy? That depends on the rest of the formula. In the formula we looked at before $\forall x \exists y P(x, y)$, x getting y as a buddy meant $P(x, y)$, so a buddy is someone that you point to. If we flip the x and the y in the binary predicate, we flip what it means to be a buddy. So if we want to satisfy $\forall x \exists y P(y, x)$, we still need to assign everyone a buddy, it's just now giving x the buddy y means that y points to x (i.e., your buddy points to you).

This even works if there are negations or other connectives involved. The formulas $\forall x \exists y \neg P(x, y)$ and $\forall x \exists y \big(C(y) \wedge P(x, y)\big)$ are still universal-existential claims. In the first case, your buddy is someone who you *don't* point to, and in the second case, your buddy has to be a circle *and* you have to point to them.

Example 3.11. Translate these formulas into English sentences, using the same definitions as before.

(a) $\forall x \exists y P(x, y)$

“Every shape has a shape that it points to,” or “All shapes have shapes that they point to,” or “For every shape, there is a shape that it points to.”

If you write “Every shape points to some shape,” I’ll give you credit for it, but sentence structures like this run the risk of becoming ambiguous. It’s not always clear whether the “some” is supposed to be inside the scope of the “every” or not. Sometimes there’s no ambiguity at all, but even in those cases, I find it easier to just avoid using “every object does a thing to some object” kinds of sentences. “Every ____ has a ____ that…” is almost always clearer.

(b) $\forall x \exists y P(y, x)$

“Every shape has a shape that points to it.”

The only thing changed here is which shape is pointing at which. It would be *wrong* to try to translate this as “Some shape points to every shape.” That is clearly an existential-universal, *not* a universal-existential, and it’s not the same thing at all.

(c) $\forall x \exists y \big(C(y) \wedge P(x, y)\big)$

“Every shape has a circle that it points to,” or “For every shape, there is a circle that it points to.”

(d) $\forall x \exists y \big(S(x) \rightarrow P(x, y)\big)$

This is an example, where a little bit of equivalence manipulation will make it easier to understand and translate: $\forall x \big(S(x) \rightarrow \exists y P(x, y)\big)$. Note that this is still a universal-existential claim, but it’s not about giving *every* shape a buddy. Instead, it’s about giving every *solid* shape a buddy.

“Every solid shape has a shape that it points to.”

(e) $\forall x \exists y \neg P(x, y)$

One very common error is to try and translate this as “Every shape does not point to some shape.” This runs into two different kinds of ambiguity problem. The most obvious ambiguity is our good old friend “Every ____ does not ____.” This translation can easily get misinterpreted as $\neg \forall x \exists P(x, y)$ (“Not every shape has a shape it points to,”) which is very different.

But there’s also a problem just with the “does not point to some shape” part. Does this mean $\exists y \neg P(x, y)$ (“there is a shape it doesn’t point to”), or does it mean $\neg \exists P(x, y)$ (“there are no shapes that it points to”)?

So “Every shape does not point to some shape,” is all around a bad idea. Fortunately, there are a couple alternate approaches. One is to stick with the “Every ____ has a ____ that…” construction:

“Every shape has a shape that it does not point to.”

> But you can also follow my earlier advice about moving the negation all the way out before translation: $\forall x \exists y \neg P(x,y) \equiv \forall x \neg \forall y P(x,y) \equiv \neg \exists x \forall y P(x,y)$. This leads to other acceptable translations:
>
> "No shape points to all shapes," or "There does not exist a shape that points to every shape."
>
> You can also try to directly translate from the halfway point of that equivalence chain: $\forall x \neg \forall y P(x,y)$. This might lead you to "Every shape does not point to all shapes," which is still ambiguous, although not as bad as the first suggestion.

If you're feeling a bit overwhelmed by that formula with the negation in it, I want to point out a useful rule of thumb for making sure you get the order of your quantifiers and negations correct.

General Principle. Always make sure that the order of the quantifier and negation symbols in the formula is the same as the order of the quantifier and negation words in the sentence. If you have $\forall$ followed by $\neg$, avoid ambiguity by translating that as "no". If you have $\neg$ followed by $\exists$, you can use "not. . . any" or "does not exist", but avoid writing "not. . . some".

So in $\forall x \exists y \neg P(x,y)$, we have the universal quantifier first, followed by an existential quantifier, and finally the negation. So any sentence that jumbles up that order, like "Every shape doesn't point to some shape," is either just wrong or ambiguous. But just getting a sentence with "every", followed by "some", followed by "not" (e.g., "Every shape has some shape that does not. . . ") gets you 95% of the way to a correct translation.

Following this guideline, you might end up translating $\forall x \neg \forall y P(x,y)$ as "Every shape doesn't point to all shapes," which is pretty darned close, and the one ambiguity can be cleared up by paraphrasing to "No shape points to all shapes."

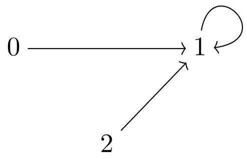
If you stick to this general principle and clean up this one ambiguity when it crops up, you'll avoid the vast majority of common translation errors involving negations.

3.4.3 Existential-Universal Claims

We talked earlier about how the model $\mathcal{M}_2$ did *not* satisfy the formula $\exists y \forall x P(x,y)$, but what kind of model *does* satisfy such a formula? Since it starts $\exists y \forall x \cdots$, this is an **existential-universal** claim. This is first and foremost an *existential* claim, so it is not about going through all the items in the universe and assigning them a buddy. This is about the existence of one special object that has a particular relationship with *everything.* If you think of a universal-existential claim as "Everyone gets a buddy," then you might think of an existential-universal claim as "There's one special, well-connected person," like that one person who happens to be friends with absolutely everybody.[40]

[40] If you're old enough to remember MySpace, think of Tom, who is automatically everyone's friend, even if they haven't added any of their own.

So to satisfy $\exists y \forall x P(x,y)$, we need to pick a single object y and make sure it is connected to *all* possible x's. If we use the same universe as before $\mathcal{U} = \{0,1,2\}$, we might pick 1 to be our one well-connected person. In this case, y being connected to x means $P(x,y)$ (i.e., that x points to y). So our well-connected person is someone that everyone points to. So we could make a model like this:



Call this model $\mathcal{M}_3$. Formally, $\mathcal{M}_3 = (\mathcal{U}, \mathcal{I}_3)$, where $\mathcal{U} = \{0,1,2\}$ and $\mathcal{I}_3(P) = \{(0,1),(1,1),(2,1)\}$.

Note that $\mathcal{M}_3$ doesn't help us disprove our earlier equivalence claim because $\mathcal{M}_3$ satisfies *both* $\exists y \forall x P(x,y)$ and $\forall x \exists y P(x,y)$. If there's one special person who gets to be everyone's buddy, then everyone *does* get a buddy (but not necessarily vice versa).

As before, we can change what it means to be "connected", but every existential-universal claim still has this same one-well-connected-person feel to it.

Example 3.12. Translate these formulas into English sentences, using the same definitions as before.

(a) $\exists y \forall x P(x,y)$

"There is a shape that all shapes point to," or "Some shape is pointed to by every shape," or "There exists at least one shape such that every shape points to that shape."

There's much less room for ambiguity here, but you do need to be careful about which shape is pointing to which shape. Also, don't start talking about "every *other* shape".

(b) $\exists x \forall y P(y,x)$

This is exactly the same as the previous formula; I just changed the names of the variables around. It's still an existential universal, and being connected to someone still means they point at you.

(c) $\exists x \forall y P(x,y)$

"There is a shape that points to all shapes," or "Some shape points to all shapes."

All we've changed here is what being "connected" means. Now being connected to someone means that you point to them.

(d) $\exists x \forall y \big(S(y) \to P(x,y)\big)$

Here we've qualified just how well-connected our special shape has to be. Instead of requiring them to be connected to *every* shape, we're just requiring that they be connected to all of the *solid* shapes. Remember that $\cdots \forall y\big(S(y) \rightarrow \cdots\big)$ by itself is translated as "... every solid shape..." So we can translate this as:

"Some shape points to all solid shapes," or "There is a shape that points to every solid shape."

Fortunately, our guiding principle about the order of quantifiers and negations doesn't apply to the order of the predicate symbols. We don't have to mention the solid shape before we mention the pointing. (Although we do need to make sure we mention "some" before "all".)

(e) $\exists x \forall y\big(C(x) \wedge P(y,x)\big)$

That $C(x)\wedge$ really doesn't need to be inside the scope of that $\forall y$ now: $\exists x \forall y\big(C(x) \wedge P(y,x)\big) \equiv \exists x\big(C(x) \wedge \forall y P(y,x)\big)$. Now we've just added an extra requirement for our special person. Remember that $\exists x\big(C(x) \wedge \cdots\big)$ by itself just means "Some circle..." or "There is a circle that..."

Don't forget that we've flipped the requirement to $P(y,x)$, so the pointing is going in the opposite direction:

"Some circle has every shape pointing to it," or "There is a circle that all shapes point to."

(f) $\exists x \forall y \neg P(x,y)$

Even with the negation, this is still an existential-universal claim. Being connected to someone now means *not* pointing at them. That makes finding a model pretty easy. Pick one special object in the universe and make sure that no matter what object you pick for a second object, the special object does *not* point to it.

Translating this one directly without ambiguity is tricky, however. Many students try to do something like "Some shape does not point to every shape," but this is definitely not correct. This sentence is the translation of $\exists x \neg \forall y P(x,y)$, which is a very weak claim, only saying that there's at least one missing arrow. This translation fails because it violates the principle I mentioned earlier: get the order of the quantifiers and the negations in the same order in the sentence and in the formula.

In this case, the order is "some", "all", "not". If your sentence doesn't have these words in the right order, it's not going to be a good translation. It's actually kind of hard to do this with the "not" appearing after both "some" and "all" because we usually like to put "not" with the verb, and we usually like to put the verb in between the subject and the object. Fortunately, we can combine the "all" followed by "not" into just "no":

"There is a shape that points to no shapes."

You can generate less direct translations by using equivalence laws to move the negation around. Applying De Morgan's Law once leads us from $\exists x \forall y \neg P(x,y)$ to the equivalent $\exists x \neg \exists y P(x,y)$. Having the negation between the quantifiers ($\exists$: "some", $\neg$: "not", $\exists$: "any") allows you to put "not" with the verb and put the verb between the subject and the object:

"Some shape doesn't point to any shapes,"

Along the same lines, you get these translations, which are acceptable, if a bit wordy:

"For some shape, there do not exist a shape that it points to," or "There is a shape that doesn't have any shapes it points to."

Remember to avoid using "some" in a negative context. "There is a shape that doesn't point to some shape," is no good. I'm not sure if it's ambiguous or just plain wrong, but it's definitely something to avoid. And "Some shape points to not some shape," is just bad English (you can't put "not" directly in front of "some" like that).

You can even pull the negation *all* the way out and then translate that: $\exists x \forall y \neg P(x,y) \equiv \exists x \neg \exists y P(x,y) \equiv \neg \forall x \exists y P(x,y)$.

"Not every shape has a shape that it points to."

3.4.4 More Than Two Alternating Quantifiers

There's a direct correlation between how many times a formula alternates between $\forall$ and $\exists$ and how hard it is for people to understand the concept being communicated. And it really has to do with alternating quantifiers, not just how many quantifiers total. A formula that starts out $\forall w \forall x \forall y \forall z \cdots$ is still pretty easy to understand ("For any four things..."). But a formula that starts out $\forall x \exists y \forall z \cdots$ is going to require some deep thought ("For every object, there is another object such that for any third object...")

Understanding a formula with, for example, 6 alternating nested quantifiers is a lot like trying to think 6 moves ahead in a game like chess or checkers. It's not something even expert players do very often. Fortunately, in the real world, we often wrap up one or more quantifiers into a single concept and give it a name. For example, you probably don't think of a statement like "x is an even" number as an existential claim, but it is! Saying "x is even" is really saying that there exists some integer n such that $x = 2n$ (i.e., x is a multiple of 2). If you spend enough time working with those concepts, you can gain intuitions that don't require breaking them down step by step every time,although you'll need to break them down when it comes time to write a proof about them.

Outside of a bonus problem or two, I won't ask you to work much with more than one alternation of quantifiers, but you will encounter them outside of this class.[41] When things start to get complicated, remember that you can always rely upon breaking up the formula piece by piece, like we did with our first

[41] One notorious example is the pumping lemma for regular languages, which you will encounter if you take Theory of Computing. The statement of that theorem has four explicit alternations in it; it's a universal-existential-universal-existential-universal claim.

problem about the difference between a universal-existential and an existential-universal claim.

3.5 Equivalence Laws for First-Order Logic

We're not going to be doing a lot of equivalence proofs for First-Order Logic, but I thought it might be good to summarize the laws we've seen so far, and to show you a few of the other important equivalence laws (even if we're not really going to use them in this class). The most important ones are the negation laws we discussed earlier:

Equivalence	Name
$\neg\,\forall x\, p(x) \equiv \exists x\, \neg\, p(x)$	universal negation (De Morgan)
$\neg\,\exists x\, p(x) \equiv \forall x\, \neg\, p(x)$	existential negation (De Morgan)

We also discussed some laws that are useful for moving quantifiers around when there are predicates that don't involve those quantifiers:

Equivalence	Name
$\forall y\big(p(x) \rightarrow r(x,y)\big) \equiv p(x) \rightarrow \forall y\, r(x,y)$	
$\exists y\big(p(x) \rightarrow r(x,y)\big) \equiv p(x) \rightarrow \exists y\, r(x,y)$	"Quantifier Movement"
$\forall y\big(p(x) \wedge r(x,y)\big) \equiv p(x) \wedge \forall y\, r(x,y)$	
$\exists y\big(p(x) \wedge r(x,y)\big) \equiv p(x) \wedge \exists y\, r(x,y)$	

Those are the ones that will be most useful for helping with translation and finding models in this class. But there are some other rules that we won't really use in this class, but that you should at least know exists. The most basic equivalence is based on the fact that the names of our variables don't matter at all. So if you change the name of a variable, the meaning of the formula doesn't change, as long as you change the name *everywhere it occurs*, and as long as the variable is a brand new variable. Obviously, you can't just change one of the x's into a y. And if there's already a y in your formula, you can't change your x's into y's. But otherwise, the names don't really matter. This rule is so basic that it's named "alpha equivalence" after the first letter of the (Greek) alphabet

Equivalence	Name
$\forall x\, p(x) \equiv \forall y\, p(y)$	"α-Equivalence" or
$\exists x\, p(x) \equiv \exists y\, p(y)$	"Variable Renaming"

We've already seen that you can't change the order of your quantifiers when the quantifiers are different (e.g., $\forall x \exists y P(x,y) \not\equiv \exists y \forall x P(x,y)$). But if the quantifiers are both universal or both existential, then you *can* reverse their order.

Equivalence	Name
$\forall x \forall y\, p(x,y) \equiv \forall y \forall x\, p(x,y)$	"Quantifier Independence"
$\exists x \exists y\, p(x,y) \equiv \exists y \exists x\, p(xy)$	

And lastly, here are a few of the rules that you could use to prove some of the "Quantifier Movement" rules we talked about earlier.

Equivalence	Name
$\forall x\big(p(x) \land q(x)\big) \equiv \forall x\, p(x) \land \forall x\, q(x)$ $\exists x\big(p(x) \lor q(x)\big) \equiv \exists x\, p(x) \lor \exists x\, q(x)$	"Distribution"
$\forall x\, p(y) \equiv p(y)$, where x is not free in $p(y)$ $\exists x\, p(y) \equiv p(y)$, where x is not free in $p(y)$	"Null Quantification"

Note that while $\forall$ distributes over $\land$ and $\exists$ distributes over $\lor$, we do *not* have distribution laws for $\forall$ over $\lor$ nor for $\exists$ over $\land$.

If you're interested in a small challenge, try to see which of the four "Quantifier Movement" rules can be proven using these more fundamental rules and which need something more powerful.

4 Informal Proof

We're not going to cover formal (or even semi-formal) Natural Deduction proofs for First-Order Logic arguments in this class. Instead, we're going to jump ahead to discuss how to use the techniques of Natural Deduction (along with a few other tools) to write proofs of claims about mathematics, set theory, and maybe even a little computer science. All the tools we learned for Natural Deduction (direct proof, proof by contradiction, proof by cases, weakening, etc.) will be useful in these less formal proofs. We'll also introduce four new-ish tools for using and proving universal and existential claims.

When you hear someone in mathematics or theoretical computer science say the word "proof", they usually mean informal proofs like the ones we'll be doing here. These proofs are only really "informal" in comparison to the formal[42] proofs that you might work with if you were working with automated theorem proving or if you were studying formal logic itself. We will be much less rigorous with these proofs than we were when we were proving claims about formulas of propositional logic, but that doesn't mean that you can get away with just anything that sounds reasonable.

In the "real" world of pure mathematics or theoretical computer science, the level of detail you provide in a proof depends dramatically on your target audience. A proof written for an undergraduate Algorithms Course textbook is going to have much more detail (with more explicitly cited rules, fewer skipped steps, and more explanations) than a proof written for the proceedings for the 14th annual Workshop on Models and Algorithms for Planning and Scheduling Problems. There's no way for me to set out a strict set of guidelines for how much detail is appropriate for any proof. The best I could say would be this:

General Principle. It should be obvious to the average reader of your proof that each step in your proof is logically valid.

If it's not immediately obvious to your reader why a step is true, then you either need to add some explanatory comments, or divide the step up into smaller steps. It's fine for your readers to be surprised by the specific steps you take, but they should *never* be unsure whether a step is actually logically valid.

[42] or to the semi-formal proofs we did earlier.

As we start writing our informal proofs in this class, I am still going to be fairly strict about the kinds of rules you can use, and how many steps you can skip. As we get further into the semester (and as our proofs get closer to the kind you are likely to see in future CS courses), I will gradually get less strict about these sorts of details. Occasionally, I will put some seemingly arbitrary limits on what you can do, but always with a good reason. Usually the reason will be that I have a particular proof technique that I want you to learn, and I want to make sure you are capable of using that technique. In those cases, I may place certain techniques and rules off-limits.

4.1 "Universal Introduction" and Proofs about Subsets

The way Natural Deduction works, there was an "introduction" and an "elimination" rule for each connective, describing how you go about *proving* a goal formula using that connective (e.g., Weakening / $\vee$-Introduction) and how you go about *using* an already assumed or already proven formula using that connective (e.g., Proof-by-Cases / $\vee$-Elimination). The same thing applies to the universal and existential quantifiers. We'll eventually need a rule for how to use existential claims ($\exists$-"Elimination"), a rule for how to prove existential claims ($\exists$-"Introduction"), a rule for how to use universal claims ($\forall$-"Elimination"), and a rule for how to prove universal claims ($\forall$-"Introduction"). Let's start by discussing how to prove a universal claim.

Imagine you're partway through a proof, you've got a couple sets named A and B, and your current goal is to prove that $A \subseteq B$. It might not seem obvious at first, but this goal is a *universal claim.* If you unpack the definition of what it means for A to be a subset of B, it becomes more obvious. In order to prove $A \subseteq B$, you would have to prove that *every* member of A is also a member of B. This is a universal claim about all the members of A. So the question becomes, how do you prove a universal claim about all the members of A?

The strategy we're going to use is very similar to the strategy we've been using for proving $\rightarrow$ formulas. Try to cast your mind back to the formal or semi-formal proofs we did for Propositional Logic formulas and remember how we went about proving a goal of the form $p \rightarrow q$. The rule we used for that was called "Direct Proof" (or "$\rightarrow$-Introduction"). Our "$\forall$-Introduction" rule is so similar that it's often also called "Direct Proof". (It's also called "$\forall$ Generalization", but I probably won't use that name in this class.)

The tools we use for universal claims are similar to the tools we use for implications because universal claims and implications are already closely related. In fact, the claim that "A is a subset of B" can be paraphrased using the symbols of First-Order Logic like this: $\forall x(x \in A \rightarrow x \in B)$. The $\rightarrow$ symbol is right there in the formula itself.

Notice that in order to use the formal language of an FOL formula, we had to introduce a new variable (x, in this case). Direct Proof for universal claims is also going to introduce a new variable, and at the same time, it's going to make an assumption about that new variable. Here's a relatively formal description of the rule:

Inference Rule (Direct Proof for Universal Claims)**.** If you start a subproof by introducing a new variable x and assuming $P(x)$ is true, and if you conclude the subproof by proving that $Q(x)$ is true, then you have proven (outside of the subproof) that $\forall x\big(P(x) \to Q(x)\big)$.

As with our other inference rules, things will make a lot more sense when we use a specific example.

Example 4.1. Write a proof of the following claim.

Claim. For all sets A, B, and C, $A \cap B \subseteq B \cup C$.

Proof.
Choose sets A, B, and C.
    Choose $x \in A \cap B$.
    So $x \in A$ and $x \in B$.
    Since $x \in B$, we know $x \in B \cup C$.
Therefore $A \cap B \subseteq B \cup C$. □

Before I get into the new stuff, let's talk about levels of formality here. This is about as informal as I'm going to allow you to get in this class for proofs about subsets. The most obvious difference here is that I haven't cited the names of any logical rules here. You certainly *can* cite the names of the logical rules you are using in each step, but I will no longer require this. But you should ALWAYS KNOW WHICH LOGICAL RULES YOU ARE USING, even if I don't require you to cite them explicitly. If you are unsure what logical rule you are using, it's probably because you are actually using a combination of several rules, or (more likely) you are relying upon general intuitions and you aren't using a logical rule at all.

The other thing you might notice is that I didn't always cite every formula that I used in each line. In particular, when the only statement you're using is exactly the thing you proved in the previous step, you can get away with just writing "so", "thus", "hence", or a similar word. These words basically mean "because of the last thing I said…" and so they're useful in these kinds of situations. So because line 2 was literally the assumption that $x \in A \cap B$ and because that's exactly the claim I used in the next step, I could get away with writing "So $x \in A$ and $x \in B$," instead of "Because $x \in A \cap B$, we know that $x \in A$ and $x \in B$."

In the following step, you'll notice that I *did* explicitly write "Since $x \in B$…", and that's because in that step, I'm only using a small part of the conclusion of the previous line. I thought it made the proof a little clearer to include that little reminder so that the reader wouldn't be surprised that I ignored the part about $x \in A$ in the next line. You could probably get away with something like "Thus $x \in B \cup C$," and that would be fine. But if you are using a statement that isn't literally the last thing you wrote, make sure to mention that fact.

The same is true for summarizing a subproof. In my head, I still summarized the subproof when I used it in that last step, but I didn't write down that summary explicitly. That's what the word "therefore" is good for. It basically means "because of all that stuff I just said...", and so it's ideal for summarizing a subproof.

Notice that each "step" in this proof is actually a combination of two steps: one involving a definition of a symbol from set theory and one involving a rule of Natural Deduction. If I were to fill in all those rule names, the proof would end up looking something like this:

Proof.

Choose sets A, B, and C.
    Choose $x \in A \cap B$.
    So $x \in A$ and $x \in B$. (Def. of $\cap$, and maybe $\wedge$-Elim)
    Since $x \in B$, we know $x \in B \cup C$. (Def. of $\cup$ and Weakening)
Therefore $A \cap B \subseteq B \cup C$. (Def. of $\subseteq$ and Dir. Pf) □

If I were to follow the full semi-formal guidelines of only one rule per step, always citing the rules, always explicitly saying which formulas I am using, and always summarizing my subproofs, the proof might look like this:

Proof.

Choose sets A, B, and C.
    Choose $x \in A \cap B$.
    Because $x \in A \cap B$, $x \in A$ and $x \in B$. (Def. of $\cap$)
    Since $x \in B$, we know that either $x \in B$ or $x \in C$. (Weakening)
    Since $x \in B$ or $x \in C$, we can conclude $x \in B \cup C$. (Def. of $\cup$)
We chose a member x of $A \cap B$ and proved that x was a member of $B \cup C$, so *every* member of $A \cap B$ is also a member of $B \cup C$. (Dir. Pf)
Because every member of $A \cap B$ is also a member of $B \cup C$, we know $A \cap B \subseteq B \cup C$. (Def. of $\subseteq$) □

The rules for $\wedge$-Elimination and $\wedge$-Introduction don't really make a whole lot of sense when we're using the English word "and" instead of the symbol $\wedge$, so that sort of thing can't really be turned into it's own separate step.

Now you do not need to give this much detail when writing proofs about sets, but I want to draw your attention to the last two steps in the proof I just wrote. Even though I'm not requiring you to write out that whole summary, **you should always be thinking it**! If you don't have that summary in your head every time you use direct proof, then there's a good chance you aren't going to do it correctly, and it will be very difficult for you to spot errors in your own proofs.

As I mentioned before, you can provide as much extra detail as you want in your informal proofs, but some students always want to know exactly how much work they *don't* have to do, so for those students, here are some guidelines about what you can get away with and what you can't get away with in your informal proofs about subsets.

Guidelines for Proofs about Subsets

- You do not need to cite the names of logical rules.

- You may combine one logical rule with one definition into a single step. You may not skip more steps than that.
- If the statement you are using is the last thing you wrote in the previous line, you may write “so”, “thus”, ”hence”, or similar instead of repeating that statement as justification for the next line.
- You may use “therefore” instead of summarizing the assumption and conclusion of the subproof you are using. But make sure you know what that summary would be, even if you don’t write it down.
- The only rules you are allowed to use for these problems are the rules of Natural Deduction (including the new ones for universal claims) and definitions of set theory symbols.

Here’s a quick reminder of the set theory definitions we’ll be using in these proofs about subsets:

Definition 4.1. $A \subseteq B$ means that every member of A is also a member of B.

Definition 4.2. $x \in A \cap B$ means that $x \in A$ and $x \in B$.

Definition 4.3. $x \in A \cup B$ means that $x \in A$ or $x \in B$.

Definition 4.4. $x \in A \setminus B$ means that $x \in A$ and $x \notin B$.

Definition 4.5. $x \in \overline{A}$ means that $x \notin A$.

The Word “Choose” and Direct Proof

The word “choose” in this proof is both introducing a new variable *and* assuming some fact about that variable. You are meant to read “Choose $x \in A \cap B$,” as “Choose a member x from the set $A \cap B$.” You could paraphrase this assumption as “Pick any member of $A \cap B$ and call it x.” Our job is to prove something about *every* member of $A \cap B$, and it’s a lot easier to do that if we give a name to some generic member of $A \cap B$ and then write the subproof as if we’re just talking about a single specific value called x. Using a new variable name allows us to carry out a sequence of complex deductions involving a generic member of $A \cap B$ without having to keep mentioning what sort of thing we’re talking about.

Some students seem to be very resistant to this kind of reasoning when it comes to statements about subsets. Instead of reasoning about one generic member of $A \cap B$ named x, they try to reason about all of the members of $A \cap B$ at the same time. You can *kind of* make that work in this problem (e.g., “Every member of $A \cap B$ is a member of both A and B, and so they must all be in B. And since every member of B is also in $B \cup C$, this means that every member of $A \cap B$ is also a member of $B \cup C$.”) But this gets harder and harder to maintain the more complex the reasoning is. (And to be honest, they usually don’t come out sounding as nice as what I just wrote.)

So don’t try to avoid learning this new Direct Proof strategy by using your own general intuitions about how $\cap$ and $\cup$ interact with subsets. Just give a name to a generic member and reason your way through to the conclusion, pretending that it's just a single object. Save the general statements about all the members of the set for when you leave the direct proof subproof.

You may have also noticed that there are actually two Direct Proofs here. There's the main direct proof with the assumption "Choose sets A, B, and C," and where we set as our main goal $A \cap B \subseteq B \cup C$. And then there's the Direct Proof subproof where we started with the assumption "Choose $x \in A \cap B$," and the goal $x \in B \cup C$." Don't forget that initial assumption for the whole proof! The claim itself might give us an obvious choice for what to name the three sets in our proof (A, B, and C), but this isn't a claim about some specific three sets. It's a claim about *any* three sets, and so we still need to introduce those variable names to the proof itself and establish that they represent any three generic sets.

4.2 "Universal Elimination"

In the same way that our Universal Introduction rule was basically a souped-up version of →-Introduction (a.k.a., Direct Proof), our Universal Elimination rule is going to be a more powerful version of →-"Elimination", also known as "Application".

Inference Rule. If you know that $\forall x\big(P(x) \to Q(x)\big)$ is true, and you have a specific object a where $P(a)$ is true, then you can apply that universal claim to a to conclude that $Q(a)$ is true.

In other words, if you know that every object with property P also has property Q, and if you know that a has property P, then you can conclude that a also has property Q. In the context of subsets, if you know $A \subseteq B$ (i.e., every member of A is also a member of B), and you know that x is a member of A, then you can conclude that $x \in B$ as well.

Let's see what that looks like in a specific example.

Claim. For any sets A, B, C, and D, if $A \cup B \subseteq C$, then $A \setminus D \subseteq C \setminus D$.

If we begin with an assumption like "Choose sets A, B, C, and D," then our main goal is "if $A \cup B \subseteq C$, then $A \setminus D \subseteq C \setminus D$." And we can prove an if-then claim with a direct proof subproof by starting a subproof assuming $A \cup B \subseteq C$ and then setting as our subgoal $A \setminus D \subseteq C \setminus D$.

Proof.

Choose sets A, B, C, and D.

  Assume $A \cup B \subseteq C$.

  ⋮

  … and hence $A \setminus D \subseteq C \setminus D$.

Therefore, if $A \cup B \subseteq C$, then $A \setminus D \subseteq C \setminus D$. □

But it's more common to absorb that second assumption into the assumption that A, B, C, and D are sets like this:

Proof.
Choose sets A, B, C, and D and assume $A \cup B \subseteq C$.

$\vdots$

... and hence $A \setminus D \subseteq C \setminus D$. □

We've set $A \setminus D \subseteq C \setminus D$ as our goal, and of course, that's a universal claim itself about all the members of $A \setminus D$. Specifically, our goal is to prove that every member of $A \setminus D$ is also a member of $C \setminus D$. So we choose a generic member of $A \setminus D$ and give it a name, and then we set out to prove that it's a member of $C \setminus D$:

Proof.
Choose sets A, B, C, and D and assume $A \cup B \subseteq C$.

> Choose $a \in A \setminus D$.
> So $a \in A$ and $a \notin D$.
> $\vdots$
> ... and hence $a \in C \setminus D$

Therefore $A \setminus D \subseteq C \setminus D$. □

Note that my assumption $a \in A \setminus D$ was based on the universal claim that was my *goal* ($A \setminus D \subseteq C \setminus D$). I also had a universal claim as one of my *assumptions* ($A \cup B \subseteq C$), but that should not influence any assumptions I make. Don't think of our $A \cup B \subseteq C$ as a rule that needs to be proven, but as a rule that *can be applied.* Specifically, if we ever find an object that is a member of $A \cup B$, then we can apply this rule to conclude that this object is also a member of C. Now it's pretty obvious that the only member we've got floating around is a, but that doesn't mean we automatically get to conclude that $a \in A \cup B$. We have to find a way to prove that ourselves, and *then* we can apply $A \cup B \subseteq C$.

Luckily, we know that $a \in A$, and we can use weakening to get $a \in A \cup B$:

Proof.
Choose sets A, B, C, and D and assume $A \cup B \subseteq C$.

> Choose $a \in A \setminus D$.
> So $a \in A$ and $a \notin D$.
> Weakening $a \in A$, we get $a \in A \cup B$.
> Thus we can apply $A \cup B \subseteq C$ to a to get $a \in C$.
> This, together with $a \notin D$ means that $a \in C \setminus D$

Therefore $A \setminus D \subseteq C \setminus D$. □

Notice that there were a couple lines (the application in line 5 and the definition of $\setminus$ in line 6) that used statements from much earlier in the proof, and in those steps, we need to explicitly mention those earlier statements, as I did in the above proof.

Let's do another example.

Claim. For all sets A, B, and C, if $A \subseteq B$, then $C \setminus A \subseteq C \setminus B$.

Proof.
Choose sets A, B, and C,and assume $A \subseteq B$.

> Choose $x \in C \setminus A$.
> So $x \in C$ and $x \notin A$.
> Uh, oh. . .
> . . . and so $x \in C \setminus B$?

Therefore $C \setminus A \subseteq C \setminus B$? □

We're kind of stuck at this point. Our goal is $x \in C \setminus B$, which means we need to prove $x \in C$ and $x \notin B$. We've already got $x \in C$, but there isn't any obvious way to prove $x \notin B$. You might be tempted to try to apply $A \subseteq B$ to x (since $x \notin A$), but application doesn't work that way. We know that every member of A is also a member of B, but that doesn't mean that every *non-member* of A is also a non-member of B. There's no reason to expect that to be true at all.

And so we're stuck. Which is a good thing, because this claim is actually false! If we somehow had "finished" the proof, then we'd be in trouble because there is no proof of this claim. If we are capable of fudging our understanding of how the rules work just to complete a "proof", then there's no point in us ever writing "proofs" at all. I've said this before, but it bears repeating:

YOU NEED TO BE ABLE TO GET STUCK WHEN WRITING PROOFS. Otherwise you'll never be able to use proof writing as a way to verify truth.

Of course, getting stuck is only the first step. Until we've properly disproved the claim, we don't know if the reason we're stuck is because the claim isn't true or if it's because we just weren't clever enough to find a valid proof. So how do we disprove a claim like this?

The same way we disprove any universal claim: with a counterexample!

Specifically, this is a universal claim about any three sets, so our counterexample must consist of three specific sets. We need to pick an A, a B, and a C in such a way that "if $A \subseteq B$, then $C \setminus A \subseteq C \setminus B$," is *false*.

Hopefully you haven't forgotten how to make an if-then statement false: make the premise ($A \subseteq B$) true, and make the conclusion ($C \setminus A \subseteq C \setminus B$) false. So in order for our example to actually be a counterexample, the set we pick for A must be a subset of the set we pick for B. We also need $C \setminus A \subseteq C \setminus B$ to be false.

How do we make $C \setminus A \subseteq C \setminus B$ false? Well that's just another universal claim, so we need another counterexample! In particular, it's a universal claim that says every member of $C \setminus A$ is also a member of $C \setminus B$. That means that our counterexample has to be a member of $C \setminus A$, but not a member of $C \setminus B$. Just to give it a name, let's call our counterexample member x.

To make sure that x is a member of $C \setminus A$, we'll need to ensure that x is in C but not in A. To make sure that x is *not* a member of $C \setminus B$, we've got two options: either make it not a member of C (impossible), or make it actually be a member of B.

Does this reasoning sound familiar? It should, because it's basically the same set-up as the attempted proof we wrote earlier. You can kind of view a proof as a carefully constructed attempt to show that a counterexample cannot exist. We got partway through the proof, constructing all the requirements for a member x of $C \setminus A$, and we found that x didn't seem to have to be a member of $C \setminus B$. It *could* have been, but it wasn't a *requirement*. That opens up the possibility of a counterexample. So this partial failed proof isn't just wasted time. It gives us a good starting point for constructing a counterexample.

So where were we? We need to pick three sets A, B, and C such that $A \subseteq B$, and we need to ensure that there is some member of C that is not a member of A but *is* a member of B.

Example 4.2. Disprove the following claim:

Claim. For all sets A, B, and C, if $A \subseteq B$, then $C \setminus A \subseteq C \setminus B$.

Let $A = \{0, 1\}$, $B = \{0, 1, 2, 3\}$, and $C = \{0, 3, 4\}$. Clearly $A \subseteq B$. $C \setminus A = \{3, 4\}$ and $C \setminus B = \{4\}$. Since $3 \in \{3, 4\}$ but $3 \notin \{4\}$, $C \setminus A$ is not a subset of $C \setminus B$.

If you're not sure whether a universal claim is true or not, you should make your best guess and either start writing a proof (if you think it's probably true) or start trying to construct a counterexample (if you think it's probably false). If you succeed at writing a proof, then congratulations, you've proven the claim is true! If you succeed at constructing a counterexample, then congratulations, you've proven the claim is false! If you get stuck in your proof or in your attempt to find a counterexample, then switch gears and try the other way. Keep switching back and forth until one of your methods succeeds and you know the final answer.

These problems involving subsets are a great way to practice proving universal claims (direct proof) and using universal claims in your proofs (application). But don't forget the other kinds of Natural Deduction subproofs, such as proof by contradiction and proof by cases.

Claim. For all sets A, B, and C, if $A \subseteq B$, then $C \setminus B \subseteq C \setminus A$.

Proof.
Choose sets A, B, and C,and assume $A \subseteq B$.

> Choose $x \in C \setminus B$.
> So $x \in C$ and $x \notin B$.
> $\vdots$
> ... and so $x \in C \setminus A$?

Therefore $C \setminus B \subseteq C \setminus A$? □

At this point, our goal is to prove that x is *not* a member of A.

Some students get here and ask me if they can apply $A \subseteq B$ backwards to x. And that's not completely unreasonable. There is a rule that says you can use $p \to q$ and $\neg q$ to derive $\neg p$. The rule is called "Modus Tollens", and it's not one of the 8 Natural Deduction rules, but it is a fairly famous rule, and in some situations (like this one), it will work just fine. But while it often seems attractive, I've found that in these subset problems, Modus Tollens can often lead you down a very confusing trail, where you end up trying to reason your way backwards through the entire proof, and the resulting proofs usually end up confusing and very hard to follow.

And you never *need* to use Modus Tollens, so in the problems we do for this class, I will usually tell you to only use the 8 Natural Deduction rules. I may even explicitly ban the use of Modus Tollens (and also De Morgan's Laws, since they're both examples of backwards reasoning) in proofs about subsets. We already have a tool (Proof by Contradiction) that can be used for proving any claim is false, and I want to make sure you know how to use that tool.

So don't use Modus Tollens as a way to avoid using Proof by Contradiction. If your goal is to prove that x is *not* a member of A, then let's just assume that x *is* a member of A and use that to derive a contradiction.

Proof.
Choose sets A, B, and C,and assume $A \subseteq B$.

> Choose $x \in C \setminus B$.
> So $x \in C$ and $x \notin B$.
>
> > Suppose towards a contradiction that $x \in A$.
> > Then we can apply $A \subseteq B$ to show that $x \in B$.
>
> But this contradicts our earlier deduction that $x \notin B$, so x must not be a member of A.
> This, together with $x \in C$ allows us to conclude that $x \in C \setminus A$.

Therefore $C \setminus B \subseteq C \setminus A$. □

Since we're not being semi-formal here, we don't have to be quite as verbose as before. But you do need to make it clear that you are writing a contradiction subproof and not a direct subproof. Here we did that by using the phrase "Suppose towards a contradiction." We also need to explicitly point out the contradiction when we finish the subproof. Note that the statement "this contradicts our earlier deduction that $x \notin B$" can appear either inside or outside of the contradiction subproof, but the conclusion that $x \notin A$ must be outside of the subproof.

Note that you should *not* use the word "choose" to introduce this contradiction subproof. When we started this contradiction subproof, we had already introduced the variable x, and "choose" only makes sense if we're introducing a new variable, not if we're adding an extra assumption about an existing variable.

Let's do one last example before we turn our attention to existential claims.

Claim. For all sets A, B, and C, if $A \subseteq C$, then $A \cup B \subseteq C \cup B$.

Proof.
Choose sets A, B, and C,and assume $A \subseteq C$.

> Choose $x \in A \cup B$.
> So $x \in A$ or $x \in B$.
> ⋮

. . . and hopefully we can prove $x \in C \cup B$.

Therefore $A \cup B \subseteq C \cup B$ □

Unlike in the previous proofs, we've found ourselves not with two facts to use, but two *possibilities* ($x \in A$ or $x \in B$). We don't know which one is true, but we do know that at least one of them is true, and so we can use proof by cases. Proof by Cases works exactly like it did when we were doing semi-formal proofs, only since we aren't being semi-formal here, we don't have to be quite as wordy.

Proof.

Choose sets A, B, and C,and assume $A \subseteq C$.

Choose $x \in A \cup B$.
So $x \in A$ or $x \in B$.
Case 1: Suppose $x \in A$.
In this case, we can apply $A \subseteq C$ to get $x \in C$.
And then we can weaken that to get $x \in C \cup B$.
Case 2: Suppose $x \in B$.
From this we can directly prove $x \in C \cup B$
In either case, we've proven $x \in C \cup B$.

Therefore $A \cup B \subseteq C \cup B$ □

We do need to write something that makes it clear that we are doing Proof by Cases (such as using the word "case" to describe the two subproofs), and of course, we have the same requirements as before: we need to already know that there are only two possibilities, and we have to prove the same statement in both cases.

4.3 Existential Claims

Just as I used subsets as our motivating example for proofs of universal claims, I've got a few different motivating examples of existential claims, pulled from simple mathematics. Take a look at the following definitions.

Definition 4.6. A number x is **even** if there exists an integer n such that $x = 2n$.

Definition 4.7. A number x is **odd** if there exists an integer n such that $x = 2n + 1$.

Definition 4.8. A number x **is divisible by** a number y if there exists an integer n such that $x = n \cdot y$.

As a shorthand for "x is divisible by y," we often write $y \mid x$. (Note the order! The bigger number comes *after* the $|$.) There are several other ways to communicate this same fact, including "x is a **multiple** of y," "y is a **factor** of x," "y **divies evenly into** x," or even just "y **divides** x." (In each of these, x is the larger number, so make sure you've got the order right. I've seen a lot of people lose points or time on tests just because they had their definitions backwards.)

Definition 4.9. A number x is **rational** if there exists integer p and q such that $x = \frac{p}{q}$ and $q \neq 0$.

In each of these examples, there's a simple-sounding claim (e.g., "x is even,") that really just shorthand for a more complex existential claim (e.g., "there is an integer such that $x = 2n$.") Specifically, these are all claims about the existence of *integers*. Note that there is no explicit requirement that the numbers we are testing be integers themselves. In the case of the rational number definition, x is rarely going to be an integer (but it might be). For the divisibility case, we will *usually* expect x and y to be integers, but they really don't have to be: since $\frac{3}{2} = 3 \cdot \frac{1}{2}$ and 3 *is* an integer, we can say that $\frac{3}{2}$ is divisible by $\frac{1}{2}$, even though neither $\frac{3}{2}$ nor $\frac{1}{2}$ is an integer. Of course, all even and odd numbers will be integers, but that's not an extra requirement; it's a *consequence* of the definitions. If you can write $x = 2n$ for some integer n, then x will be an integer. No need to mention that explicitly in your proofs.

Since we'll be doing so many problems involving integers, it's worth it to pause and talk about what sorts of facts we know about math that will be permitted in proofs in this class. When we were writing proofs about subsets and set operations, I limited the facts you could use to just the definitions (and the logical rules, of course). I'm not going to be *quite* so strict with our proofs about numbers, but I've still got a few restrictions I would like you to stick to.

When it comes to issues about even and odd numbers, divisibility, or rational numbers, THE ONLY FACTS YOU MAY USE ARE THE DEFINITIONS ABOVE. While it's true that adding two even numbers always results in another even number, this is not a fact you are allowed to use in your proofs. Similarly, you can't assume that adding two rational numbers will result in another rational number, that squaring an even number results in another even number, etc.

Other than that, you can use any simple rule from high school algebra that you want, as long as it's something you can expect your target audience to already know. For this class, you can assume that your target audience is your fellow classmates. I know that this is a bit of a fuzzy guideline, but that is the nature of informal proofs.

So for example, you should feel free to use facts that can be easily checked with a calculator ($7 \cdot 12 = 84$), rules of arithmetic ($x + y = y + x$), basic algebraic manipulations ($\frac{a}{b} + \frac{c}{d} = \frac{ad+bc}{bd}$, $(x+3)^2 = x^2 + 6x + 9$), and similarly well-known tools. One thing that is worth mentioning are **closure** properties for integers.

Fact. The integers are **closed** under addition, subtraction, and multiplication. This means that for any two integers n and m, you can conclude that $n + m$, $n - m$, and $n \cdot m$ are also integers.

I mention these rules specifically by name for two reasons. First of all, I want to make sure you are aware of this usage of the word "closed". Secondly, I want to make it clear that I am *not* allowing you to use any closure properties for even numbers, odd numbers, numbers divisible by a specific number, or rational numbers. There *are* such closure properties, but those are the kinds of claims we're going to be *proving* in this class, so you should not be *using* those properties, especially not as a way to avoid working with existential claims.

4.3.1 Existential "Introduction"

With those preliminaries out of the way, let's take a look at how we'll work with existential claims in our proofs. First of all, we're going to need an "Existential Introduction" rule that will allow us to prove that an existential claim is true. Only you already know how to this: give an example! So the only real new thing I have to teach you about this is how to express thse ideas in a proof.

Claim. The number 27 is odd.

Now this is a really simple claim, and the proof is going to be absurdly simple. But I want to use it just to demonstrate the language we use when we want to prove an existential claim in a proof.
Proof.
Let $n = 13$.
Clearly n is an integer.
$2n + 1 = 2 \cdot 13 + 1 = 27$
Hence 27 is odd. □

I've divided the proof into separate lines, but this isn't really necessary. You could just as easily write this proof in one paragraph:

Proof. Let $n = 13$. Clearly n is an integer. $2n + 1 = 2 \cdot 13 + 1 = 27$, and hence 27 is odd. □

We needed to prove the existence of an integer with specific properties, and we did this by giving an example of such an integer (13, in this case). When giving an example to prove an existential claim, it's important to draw the reader's attention to the example and then to verify that it has all the required properties. In this particular proof, we gave the example a name n as a way of drawing attention to the number. Of course, you don't have to give it a new name if you don't want to. The number 13 already has a perfectly good name: "13"!

But you still need to draw attention to the example that you want your reader to be thinking about. If you just started off with $27 = 2 \cdot 13 + 1$, the reader isn't going to know why that equation is relevant. It's supposed to explain how 27 and 13 are related to each other, but if they aren't already thinking about 13, it might not be obvious to them what the point of the equation is. Here are a couple more simple mini-proofs, just to give you an idea of what your options are.

Claim. 2001 is divisible by 23.

Proof. Consider the integer 87. Since $23 \cdot 87 = 2001$, we know that $23 \mid 2001$. □

Here, I used the word "consider" as a way to introduce the value 87 as the example integer we were using to prove divisibility.

Claim. The number 0.15 is rational.

Proof. Obviously, $15 \in \mathbb{Z}$, $100 \in \mathbb{Z}$, and $100 \neq 0$. So because $0.15 = \frac{15}{100}$, $0.15 \in \mathbb{Q}$. □

Notice that in these proofs, many of the requirements that we needed to establish didn't really require any argumentation. It's obvious that 15 is an integer and that 100 is not the same number as 0. But because they are requirements for the definition of a rational number, I STILL HAD TO MENTION THEM IN THE PROOF. This is why you see words like "clearly" or "obviously" in proofs. We use these words when we need to mention *that* a particular fact is true, but we don't need to explain *why* that fact is true.

Of course, we usually aren't called to prove bare existential claims. More likely, we are going to be using this strategy as part of proving a universal-existential claim or an existential-universal claim, such as:

Claim. There is an integer that is a factor of every integer.

This is an existential claim, so we're going to start off the proof just by giving the example. But since it's an existential-*universal* claim, once we've given the example, our proof that the example works will be a little more complicated than just checking off the requirements and using a calculator. But fortunately, we already know how to prove a universal claim by using direct proof.

Proof.

Consider the integer 1.

  Choose an integer n.

  Clearly $n = n \cdot 1$, and we already established that $n \in \mathbb{Z}$.

  Hence n divides evenly into n.

So every integer is divisible by 1. □

Things get slightly more interesting when we look at universal-existential claims, like this one:

Claim. 10 times any integer is even.

This is a universal claim, so we can begin by setting up a direct proof.

Proof.

  Choose an integer n.

  $\vdots$

  $\cdots$, and hence $10n$ is even.

Therefore, 10 times any integer is even. □

The goal of our direct proof is the existential claim that $10n$ is even. According to the definition of "even", we have to give an example of an integers m such that $10n = 2m$. But of course, we can't find a *specific* integer m like 7 or 208 because we don't know exactly which integer n actually is.

But the whole point of using direct proof here is to give the name n to a generic integer, so that we can *act as if* n actually *is* a specific number. And if

we treat n like a specific number, we can use it to define other specific numbers, like $n+1$, n^2, or (in this case) $5n$:
Proof.

Choose an integer n.
Let $m = 5n$. Since 5 and n are integers, so is m.
Because $10n = 2 \cdot 5n = 2m$, we know that $10n$ is even.
Therefore, 10 times any integer is even. □

(Honestly, you could probably leave out that last line, and the proof would be just as clear.)

Note that in order to prove that m was an integer, I had to use the fact that the integers are closed under multiplication. I didn't specifically cite the rule, although I could have done so by writing something like "$5n \in \mathbb{Z}$ because the integers are closed under multiplication." The closure properties of the integers are definitely simple enough that you don't need to cite the rules by name. But it does need to be clear to your reader that this you are using this fact, which is why I explicitly mentioned that 5 was an integer, and why I repeated the fact that n was an integer. Those two facts were necessary to show that $m = 5n$ was an integer, so even though one was obvious and one was mentioned earlier, I still needed to repeat them at the appropriate time.

4.3.2 Existential "Elimination"

So the proofs we've done so far with existential claims are almost stupidly simplistic. Things don't really get interesting until we start talking about how to *use* existential claims. We know what to do when one of our *goals* is an existential claim: give an example. But what do we do when one of our *assumptions* is an existential claim?

Claim. The sum of any two odd numbers is even.

This is clearly a universal claim, so we can set up our direct proof right away, with appropriate assumptions and goals.
Proof.

Choose two odd numbers n and m.
$\vdots$
$\cdots$, and so $n+m$ is even.

□

The goal of this direct proof is to show that $n+m$ is even, which is an existential claim, which we again know how to deal with. We need to come up with some integer, so that two times that integer is equal to $n+m$. So hopefully we can get our proof to look something like:
Proof.

> Choose two odd numbers n and m.
>
> $\vdots$
>
> $n + m = 2 \cdot \langle \textit{some integer} \rangle$ and $\langle \textit{some integer} \rangle$ is an integer, so $n + m$ is even.

□

So we just need to find out what value $\langle \textit{some integer} \rangle$ should be, and then show that it has the required properties, and we're good to go. But how are we going to find such a number?

We're definitely going to have to use the fact that n and m are odd to help us out here. From the definition of "odd", we know that there is going to be some integer (let's call it k) so that $n = 2k + 1$ (and another such integer for m). I couldn't use the same variable as in the definition of "odd" because we were already using n for something else. And that right there is the only trick to existential "elimination".

All you have to do to use an existential claim is to pick a new variable name to stand in for the value that you know exists. We may not know exactly *which* integer k is, but we know that there must be *some* such integer (because we know n is odd), so we might as well give it a name! The only thing we have to be careful about is to make sure that the name we choose isn't already in use for some other purpose.

So let's do that for our example:

Proof.

> Choose two odd numbers n and m.
> Since n is odd, there must be an integer k such that $n = 2k + 1$.
> Similarly, since m is odd, $m = 2j + 1$ for some integer j.
>
> $\vdots$

□

Now we have enough names that we can say something interesting (and hopefully useful) about $n + m$:

$$\begin{aligned} n + m &= (2k + 1) + (2j + 1) \\ &= 2k + 2j + 2 \end{aligned}$$

We now have $n + m$ written as three numbers added together: $2k + 2j + 2$, but what we really need is $n + m = 2 \cdot \langle \textit{some integer} \rangle$. How do we rewrite an addition expression as a multiplication expression? We factor it!

This gives us $n + m = 2(k + j + 1)$, which is exactly what we wanted. Or rather, if we establish that $k + j + 1$ is an integer, *then* we have exactly what we wanted.

Proof.

Choose two odd numbers n and m.
Since n is odd, there must be an integer k such that $n = 2k + 1$.
Similarly, since m is odd, $m = 2j + 1$ for some integer j.

$$\begin{aligned} n + m &= (2k + 1) + (2j + 1) \\ &= 2k + 2j + 2 \\ &= 2(k + j + 1) \end{aligned}$$

Since k, j, and 1 are all integers, so is $k + j + 1$.
So since $n + m = 2(k + j + 1)$, $n + m$ must be even.

□

This process of giving a new name to a value we know exists is called "Existential Elimination" or sometimes "Existential Instantiation".[43] If you've followed all of the other kinds of rules we've had so far, it shouldn't be too difficult to pick up. There's really only one common mistake that I see with existential instantiation, and that is when you accidentally give the same name to the two integers that we know exist. If you used both $n = 2k + 1$ and $m = 2k + 1$, your proof would be invalid, so always be careful that your new variable is truly new.

I want to draw your attention to the sentence "Since k, j, and 1 are all integers, so is $k + j + 1$." This sentence serves two purposes: it verifies that $k + j + 1$ is an integer, but it also draws attention to $k + j + 1$ as a single specific number, which is what we need so that our readers understand what we are doing, i.e., we are proving that *there exists* such an integer. This is a kind of bottom-up sort of proof, where we derive the formula for the example we want from facts we already know, only drawing attention to the example as an integer at the very end. You can also write this proof in a sort of top-down fashion, defining the example immediately, and then showing that it solves the necessary equation:
Proof.

Choose two odd numbers n and m.
Since n is odd, there must be an integer k such that $n = 2k + 1$.
Similarly, since m is odd, $m = 2j + 1$ for some integer j.
Let $N = k + j + 1$. $N \in \mathbb{Z}$ because k, j, and 1 are all integers.

$$\begin{aligned} 2N &= 2(k + j + 1) \\ &= 2k + 2j + 2 \\ &= (2k + 1) + (2j + 1) \\ &= n + m \end{aligned}$$

Therefore $n + m$ must be even.

[43] If you want to get really technical about how the rule works, we would actually be creating a subproof for the new variable k. And we wouldn't be able to leave the subproof until we proved some statement that didn't mention k at all (like "$n + m$ is even.") And once we had that statement, we could bring it out of the subproof, in the same way that proof by cases works. In fact, you can think of existential instantiation as doing infinitely many "cases", but just doing them all at the same time.

□

Let's do one last example just for practice.

Claim. The rational numbers are closed under multiplication.

Proof.

Choose rational numbers x and y.
So there exist integers p_1, p_2, q_1, and q_2 with $x = \frac{p_1}{q_1}$, $y = \frac{p_2}{q_2}$, $q_1 \neq 0$, and $q_2 \neq 0$.
$x \cdot y = \frac{p_1}{q_1} \cdot \frac{p_2}{q_2} = \frac{p_1 \cdot p_2}{q_1 \cdot q_2}$
Since p_1, p_2, q_1, and q_2 are all integers, $p_1 \cdot p_2$ and $q_1 \cdot q_2$ are also integers.
$q_1 \cdot q_2$ can't be zero because neither q_1 nor q_2 is zero.
Therefore $x \cdot y$ is rational.

□

This proof, like many proofs, is a mishmash of lots of different techniques that we've seen in this class. The whole structure is a direct proof of a universal claim. Inside of that direct proof, we're giving an example to prove an existential claim. In order to construct our example, we're using our new existential instantiation rule to give names to the values we know exist. And finally, to demonstrate that our example has the right properties, we're using a chain proof.

4.3.3 Negated Existential Claims

Consider the following claim:

Claim. For any integer n, if n is divisible by 3, then $(n + 1)^2$ is *not* divisible by 3.

We can get started on the proof just as before:
Proof.

Choose an integer n and assume that $3 \mid n$.
So there exists an integer k such that $n = 3k$.
$\vdots$
$\ldots$, and therefore $3 \nmid (n + 1)^2$.

□

Our goal is no longer an existential claim. We're not trying to prove that $(n+1)^2 = 3 \cdot \langle \textit{some integer} \rangle$. In fact, we're trying to prove that *no such integer exists*! So we definitely should not be trying to find an example. Instead, we are going to rely upon the most general method for proving that a claim is *not* true: proof by contradiction.

There are certain situations in which there are better approaches to proving negated claims than proof by contradiction. Proving that a *universal* claim is false is more easily done by giving a counterexample. But proving that an *existential* claim (like divisibility) is false is an excellent candidate for proof by contradiction.

Proof.
Choose an integer n and assume that $3 \mid n$.
So there exists an integer k such that $n = 3k$.
    Suppose towards a contradiction that $3 \mid (n+1)^2$.
    So there exists an integer m such that $(n+1)^2 = 3m$.
    $\vdots$
    $\ldots$, which is (hopefully) impossible!
Therefore $3 \nmid (n+1)^2$. □

Our subgoal is to prove some kind of contradiction. In a problem about numbers like this, there are many, many different kinds of contradictions one could prove. Perhaps we might end up proving that $1 = 2$ (obviously false) or that $\frac{1}{2} \in \mathbb{Z}$ (again, obviously false). Or we might end up proving two new claims that contradict each other. In many problems, there will be a natural choice for what contradiction we should seek out. But this particular problem is tricky because there really isn't an obvious choice right at the beginning. So let's just play around with algebra and see what we can find.

What do we have to work with? Well we've got two equations: $n = 3k$ and $(n+1)^2 = 3m$, so it's a good bet that we'll have to combine those equations somehow. Substitution is a good general strategy for combining two equations into a single statement, and since one of these equations is solved for n already, we might as well do that substitution:

$$\begin{aligned} 3m &= (n+1)^2 \\ &= (3k+1)^2 \end{aligned}$$

This doesn't obviously get us anywhere, unfortunately. But perhaps if we distribute the polynomial, things might look different.[44] Remember that we're looking for a contradiction, so we should keep our eyes open for anything that looks a bit off.

$$\begin{aligned} 3m &= (n+1)^2 \\ &= (3k+1)^2 \\ &= (3k)^2 + 2(3k) + 1 \\ &= 9k^2 + 6k + 1 \end{aligned}$$

If you're paying attention, this might start to look a little wrong to you. (This is a good sign, since we're trying to find a contradiction!) On the left

[44] Note that this isn't actually *simplification*. Any polynomial can be written either in factored form or in distributed form. Both forms are simple. And sometimes there's an obvious choice for which version is most promising for your proof. But in this case, we're only distributing because it's an obvious thing that we can do. We're really just hoping for something to start looking fishy.

hand side, we have $3m$, an obvious multiple of 3. On the right hand side, we're adding two multiples of 3 together and then adding 1. If you've got a good intuition about multiples of 3, you may have realized that this shouldn't result in a multiple of 3. Our intuition is not a proof, of course, but that nagging feeling that something is wrong is a sign that we're heading in the right direction. Let's try to manipulate this a bit further in the hopes of getting a form where the impossibility is more obvious. In particular, our intuitions tell us that $9k^2$ and $6k$ are divisible by 3 and so their sum is also divisible by 3. We can make that more explicit in the equation:

$$\begin{aligned}3m &= (n+1)^2\\ &= (3k+1)^2\\ &= (3k)^2 + 2(3k) + 1\\ &= 9k^2 + 6k + 1\\ &= 3(3k^2 + 2k) + 1\end{aligned}$$

We're definitely getting closer here. Now the right hand side is a multiple of 3 plus 1, which really shouldn't be equal to a multiple of 3. Depending on your audience, this might be enough of a contradiction. For this class, I've asked you only stick to the definition of divisibility and not rely upon your more general intuitions about divisibility, and we can definitely do better, so let's do that. Here's a nifty trick for turning a statement about divisibility into a statement about integers: divide both sides by 3:

$$m = (3k^2 + 2k) + \frac{1}{3}$$

Now things are looking even better. We have an integer $3k^2 + 1$ plus a non-integer $\frac{1}{3}$, and the result is equal to an integer m. You could probably get a pretty good contradiction out of this if you phrased things carefully, but we can even go one step further and solve for $\frac{1}{3}$:

$$\frac{1}{3} = m - 3k^2 - 2k$$

And this is definitely a contradiction. We can show that the right hand side is an integer (since the integers are closed under addition, multiplication, and subtraction and since m, k, 2 and 3 are integers), and the left hand side is clearly *not* an integer. Here's what that might look like in the proof:

Proof.

Choose an integer n and assume that $3 \mid n$.

So there exists an integer k such that $n = 3k$.

Suppose towards a contradiction that $3 \mid (n+1)^2$.

So there exists an integer m such that $(n+1)^2 = 3m$.

$$\begin{aligned}
3m &= (n+1)^2 \\
&= (3k+1)^2 \\
&= (3k)^2 + 2(3k) + 1 \\
&= 9k^2 + 6k + 1 \\
&= 3(3k^2 + 2k) + 1 \\
m &= (3k^2 + 2k) + \frac{1}{3} \\
\frac{1}{3} &= m - 3k^2 - 2k
\end{aligned}$$

Since m, k, 2, and 3 are integers, so is $m - 3k^2 - 2k$. But this is impossible because $\frac{1}{3}$ is clearly *not* an integer.

Therefore $3 \nmid (n+1)^2$. □

Let's do one more example: one where one of our *assumptions* is a negated existential claim.

Claim. The product of a rational number and an irrational number is always irrational.

Proof.

Choose a rational number x and an irrational number y.
Since x is rational, there exist integers p and q such that $x = \frac{p}{q}$ and $q \neq 0$.

$\vdots$

□

I want to remind you of what's going on in that last line I wrote. On the one hand, it looks an awful lot like I just unpacked the definition of what it means for x to be a rational number. But it's a bit more subtle than that. In the definition of what it means for x to be rational, the integers p and q only have meaning *inside that definition.* To make a programming metaphor, they're like local variables in a one-line function definition. I am free to use p and q again in other definitions and there's no reason to think that the p in one definition has any connection to the p in other definitions.

But when I am using an existential claim in a proof like this, things are different. Even though the sentence is exactly the same, the p and q that I introduced in the proof are going to be used *throughout the proof.* It makes sense to do this because if we commit to our assumption that x is rational, then we know that p and q must exist (even if we don't know exactly what integers they are). So it makes sense to give them names and then use those names to refer to those integers for the remainder of the proof.

This is a subtle distinction, but it's important to understand the definitions so that you don't fall into the trap of writing as the next line of the proof:

> Since y is not rational, there do not exist integers m and n such that $y = \frac{m}{n}$ and $n \neq 0$.

To be clear: the above sentence is *true*, but you probably should not include it in your proof at this point. You *definitely* should not include it in your proof and then continue to use the variables m and n. That wouldn't make any sense at all. The whole point of saying that y is *not* rational is to say that *there is no* m and n with those properties. If we were to give those non-existent numbers names, we'd be claiming that they *do* exist, and that's a problem.

So if we're not going to introduce variables to stand in for those values, what can we do with a negated existential assumption like this? Well, not much, actually. Certainly we can't do anything with the fact that y is irrational *right now.* We're actually going to treat them kind of like how we treat universal assumptions: we're just going to stick them in our back pocket and hope that they turn out to be useful later. In particular, negated claims like these are particularly useful if we ever find ourselves in a proof by contradiction later on.

So let's just stick the fact that y is rational into our tool belt and hope that we can use it later. In the meantime, we can turn our attention to our goal, which is to prove that xy is not rational. As with our last proof, we have a goal which is to prove some claim *false*, so we can use proof by contradiction.

Proof.
Choose a rational number x and an irrational number y.
Since x is rational, there exist integers p and q such that $x = \frac{p}{q}$ and $q \neq 0$.

> Suppose that xy is rational. (Will show a contradiction.)
> So there must exist integers j and k with $xy = \frac{j}{k}$ and $k \neq 0$.
>
> $\vdots$
>
> . . . , which (hopefully) is impossible.

Therefore, xy is not rational. □

We've set up a vague goal of "any contradiction" here, and we could just try stuff and hope to stumble across a contradiction, as we did last time. But remember that we stuck a tool into our toolbelt. It's even a tool that's particularly useful for finding contradictions. Remember that we know that y is *not rational.* So if we could somehow prove that y *were* rational, that would definitely be a contradiction. That actually gives us a nice concrete goal:

$$y = \frac{\langle \textit{some integer} \rangle}{\langle \textit{some non-zero integer} \rangle}$$

How do we achieve this goal? Well we can start by combining the two equations we already have $x = \frac{p}{q}$ and $xy = \frac{j}{k}$:

$$\frac{p}{q} \cdot y = \frac{j}{k}$$

And then we can solve for y:

$$y = \frac{j}{k} \cdot \frac{q}{p}$$

Now that we have $y = \langle \textit{something} \rangle$, we can focus on taking that $\langle \textit{something} \rangle$ and trying to write it in the form $\frac{\langle \textit{some integer} \rangle}{\langle \textit{some non-zero integer} \rangle}$, which is pretty easy in our case:

$$y = \frac{j}{k} \cdot \frac{q}{p} = \frac{jq}{kp}$$

Looking good! There are just a few more details to clean up. The first is to show that jq and kp are actually integers. That's pretty easy to do because j, k, p, and q are all also integers (and we know that the integers are closed under multiplication).

So the only thing left to do is to prove that kp is not zero. Fortunately we already know that $k \neq 0$ and $q \neq 0$... Wait a minute... Something's not right. We need to have $p \neq 0$, but we don't have that. Did we make a mistake?

Our reasoning is sound, unfortunately. We definitely need p to be non-zero if we're going to prove that kp is not zero, and there's simply no way to prove that given the assumptions we have.

So let's pause and think about what it would mean if p *were* zero. Well, since $x = \frac{p}{q}$, if $p = 0$, then that means that x is *also* equal to zero, which is certainly possible, since zero is a rational number. And if $x = 0$, then that means that x times y is also 0, no matter what irrational number we pick for y.

That actually means that the claim is *false*! We can prove that with a counterexample: 0 is a rational number, and $\sqrt{2}$ is a rational number, and their product $0 \cdot \sqrt{2}$ is 0, which is also a rational number.

So there really wasn't any hope of proving this claim in the first place. Note that we never would have discovered this if we hadn't gotten stuck in the proof. If we'd tried to force the completion of the proof by adjusting/destroying our understanding of how the rules work, or by skipping over details that we didn't fully understand, we would have a garbage proof on our hands.

I've said it many times before, and I will say it again: In order for proofs to be useful, YOU HAVE TO BE ABLE TO GET STUCK.

Fortunately, this was just one edge case counterexample, not a fundamental flaw in our argument. So while the original claim isn't true, we *can* write a proof of a slightly modified claim:

Claim. The product of a *non-zero* rational number and an irrational number is always irrational.

Proof.

Choose a *non-zero* rational number x and an irrational number y.

Since x is rational, there exist integers p and q such that $x = \frac{p}{q}$ and $q \neq 0$.

Because $x \neq 0$, we know $p \neq 0$.

Suppose that xy is rational. (Will show a contradiction.)

So there must exist integers j and k with $xy = \frac{j}{k}$ and $k \neq 0$.

$$\begin{aligned}\frac{p}{q}\cdot y &= \frac{j}{k}\\ y &= \frac{j}{k}\cdot\frac{q}{p}\\ &= \frac{jq}{kp}\end{aligned}$$

Since j, k, p, and q are integers, so are jq and kp.

Since neither k nor p is zero, kp isn't zero either.

Therefore y is a rational number.

But y can't be rational because we assumed it was irrational earlier, and Therefore, xy is not rational. □

5 Relations

There's a strong connection between *properties* (e.g, whether a number is even, whether a set is finite, whether a formula is satisfiable, whether a string is alphanumeric,...) and sets (e.g., the set of all even numbers, the set of all finite sets, the set of all satisfiable formulas, the set of all alphanumeric strings,...). Any property can be used to define a set (using set builder notation) and every set defines a property (the property of being a member of that set). You can think of a set as a formal way of representing a property.

These properties are represented in first-order logic by unary predicates (e.g., $E(x)$, $F(x)$,...). This raises a natural question: "What about binary predicates (e.g., $D(x, y)$, $S(x, y)$,...)?"

We use the term **relation** to talk about these "properties" that exist *between* objects. Examples include whether one number is divisible by another, whether one set is a subset of another, whether one formula is equivalent to another, whether two strings are anagrams of each other,.... What is the formal set-theoretic definition of a **relation**?

Well, to get a property from a set, we can use set-builder notation. For example, the property of being an even number allows us to define the set $\{n \mid n \text{ is even }\}$. This set's members are all of the even numbers.

For relations, it's not about which things have a property. It's about which *pairs* of things have the relationship. So the set that represents divisibility can't have single numbers as members, but instead should have *pairs* of numbers as its members. So the pair $(24, 6)$ would be a member of the divisibility relation, but the pair $(24, 5)$ would not.

And this is exactly how we will define a relation: a set of pairs.

Of course we should probably be specific about what we mean by the word "**pair**". We can't just use a two-item list because relations are often (but not always) directional. 24 is divisible by 6, but 6 is not divisible by 24. So we need a different kind of object: one that is like a set, but where the order matters.

5.1 Ordered n-tuples

Sets are very useful when you want to talk about groups of things without distinguishing special positions for each member in the group. So sets are good when you want to talk about which numbers have a particular property, or which values can possibly be returned by a particular function in a computer program, or which students are members of a particular club. But if you wanted to talk about the coordinates of a point on the graph of an equation, or the list of arguments being passed to a function, or a queue of people waiting in line, then a set might not be the best tool for the job.

In all these examples, the order in which things occur is important. Consider the point which is 5 units to the right of the origin and 3 units down. If we tried to express this as a set, we might end up with $\{5, -3\}$. But this is exactly the same as the set $\{-3, 5\}$, so there'd be no way to distinguish between the point at 5 right, 3 down and the point at 3 left, 5 up. You'd run into even more trouble if you tried to describe the point that's 7 right and 7 up.

If you remember your algebra, you'll know that mathematicians have a different tool for describing points like this: the **ordered pair**. An **ordered pair** (or **ordered couple**) consists of two **coordinates** (or **components**), with distinct positions. We write an ordered pair the same way as we write a set, only we use (round parentheses)[45], the order matters, and you're allowed to reuse the same value for different coordinates. We often drop the "ordered" part, and just say "pair" or "couple."

So we can model the point at 5 right, 3 down with the ordered pair $(5, -3)$ and the point at 3 left, 5 up with the pair $(-3, 5)$, and those two pairs are considered different things.

Of course, we're not limited to just two coordinates. We can talk about the ordered **triple** $(1, -2, 17)$, or the ordered **quadruple** $(5, 2, 5, -2)$, or so on.

number of coordinates	English word
2	pair / couple
3	triple
4	quadruple
5	quintuple
6	sextuple
7	septuple
8	octuple
⋮	⋮
n	n-tuple

This naming scheme is fine as long as we never change the number of coordinates (and as long as we remember our Latin numerical prefixes). And surprisingly often, we won't need to do change the number of coordinates. In any case, the generic term for these kinds of objects is **tuple**.[46] We'll mostly be

[45] In some situations, you may see ⟨angle brackets⟩ as well.

[46] "Tuple" is sometime pronounced /tuːpəl/ so that the first syllable sounds the same as the first syllable of "toupee" and it's sometimes pronounced /tʌpəl/, so that the first syllable is the same as the first syllable in "Tupperware." Some people even pronounce it /tjuːpəl/, so

working with ordered pairs in this class, but you should be aware of the more general concept.

Okay, so now we have a formal way of talking about pairs of things. Since our relations are going to be sets of pairs, it's worthwhile to do a few examples that makes sure we don't get too confused about them.

Example 5.1. True or False:

(a) $(1, 2) = \{1, 2\}$

False. One is a pair, the other a set.

(b) $(1, 2) = (2, 1)$

False. Order matters for ordered pairs.

(c) $(1, 1, 2) = (1, 2)$

False. One is a triple and the other is a pair.

(d) $\big\{\{1, 2\}, \{2, 3\}, \{3, 1\}\big\} = \big\{(1, 2), (2, 3), (3, 1)\big\}$

False. $(1, 2)$ is a member of the set on the right, but not the left. (The set on the left is a set of *sets* of numbers, but the set on the right is a set of *pairs* of numbers.)

(e) $\big\{(1, 2), (2, 3), (3, 1)\big\} = \big\{(2, 3), (1, 2), (3, 1)\big\}$

True. The two sets have the same three members. (Order doesn't matter in listing the members of a set.)

(f) $\big\{(1, 2), (2, 3), (3, 1)\big\} = \big\{(2, 1), (2, 3), (3, 1)\big\}$

False. $(1, 2)$ is a member of the set on the left, but not the right. (Order *does* matter in listing the coordinates of an ordered pair.)

5.2 Defining Relations

Definition 5.1. A **relation from** a set A **to** a set B is a set of ordered pairs where the first coordinates are all members of A, and the second coordinates are all members of B. We call A the **domain** of the relation, and we call B the **codomain** of the relation. When the domain and codomain are both the same set A, we say that it is a **relation on** A.

As you've just seen, one way to define a relation is to use set-list notation and just list off all of the pairs of things that you want to be related. Any pair you don't want to be related is simply not listed.

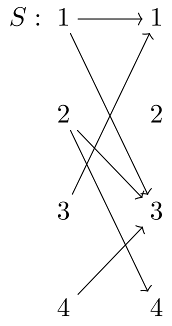
Example 5.2. Let $A = \{1, 2, 3, 4\}$ and $B = \{\triangle, \square, \bigcirc\}$. Define a relation R from A to B as follows: $R = \{(1, \square), (1, \bigcirc), (2, \square), (3, \bigcirc)\}$. Let S be a relation on A defined by $\{(1, 1), (1, 3), (2, 3), (2, 4), (3, 1), (4, 3)\}$.

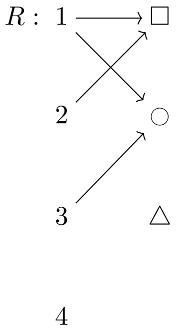
that it sounds like "pupil," only starting with a "t" instead of a "p."

Once you have a relation, there are several ways you can describe whether or not a particular pair of values is related. As an example, here are several ways to communicate the same fact:

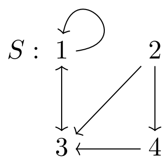
- Using set-theory notation: "$(2, 3) \in S$".
- In English: "S relates 2 to 3," or "2 is related to 3 (by S)."
- Prefix notation (this is what First-Order Logic does): "$S(2, 3)$".
- Infix notation (this is usually used when the relation is given a symbol for a name, but is occasionally used when the relation has a letter for a name): "2 S 3"

An alternate way of communicating the same information as a list of pairs is to draw the elements of the domain and codomain, and draw arrows between the pairs that we want to be related. So here are some alternate definitions of the relations R and S defined above:





When the domain and the codomain are the same, it's common not to separate the domain and codomain, resulting in a graph like this:



These kinds of pictures with nodes connected by directional edges (arrows) are called **directed graphs** and they can be used to define relations in much the same way as set-list notation.

Of course, if we only use set-list notation or directed graphs to define our relations, then we're limited to small, finite relations. This is sometimes useful. In the honors section, we've been using small, finite relations like these to build our First-Order Logic models. But if our relations are going to be infinite, or even just large, we're going to need to use set-builder notation to define our relations.

Example 5.3. Define a relation L from the set of text strings to $\mathbb{N}$ by $L = \{(s, n) \mid s \text{ has length } n\}$. Let G be a relation on $\mathbb{R}$ defined by $G = \{(x, y) \mid x \geq y\}$.

If we want to know whether $G(7,3)$ is true, we just check to see if $(7,3)$ is a member of G. According to the set-builder definition, $(7,3) \in G$ iff we can write $(7,3)$ in the form (x, y), where $x \geq y$. Clearly, if $x = 7$ and $y = 3$, we've met the requirement ($x \geq y$), so $G(7,3)$ is true (i.e., G relates 7 to 3). On the other hand $G(4,5)$ is *false* because $(4,5) \notin G$ (because when $x = 5$ and $y = 4$, $x \ngeq y$).

For similar reasons, we can say that L relates `"foobar"` to 6 (in other words, $L($`"foobar"`$, 6)$), but it doesn't relate `"foobar"` to 3 (in other words, $\neg\, L($`"foobar`$, 3)$).

There are a few trivial or nearly trivial relations that you should know about. The first is the **empty relation**. If a relation is just a set where all of its members are pairs, then technically speaking, the empty set $\varnothing$ is a relation. It's a stupid relation because it doesn't relate anything to anything but it is technically a relation.

Another relation that's not quite as trivial as the empty relation is the **identity relation** on a set. If you have a set A, then the identity relation $I = \{(x, y) \mid x = y\}$ relates every object to itself and not to anything else. So the identity relation on $\mathbb{Z}$ relates 5 to 5, but doesn't relate 5 to anything else. It also relates -17.2 to -17.2, but not to anything else. It's not quite as silly as the empty relation, but it is pretty simplistic.

5.3 The Cartesian Product

At the other end of the spectrum from the empty relation is the *total relation* which relates absolutely everything in the domain to absolutely everything in the codomain. You could express the total relation on $\mathbb{Z}$ using set builder notation like this: $T = \{(x, y) \mid x \in \mathbb{Z} \wedge y \in \mathbb{Z}\}$. So $T(1, 7)$ would be true, but so would $T(1, 1)$, $T(1, -5)$, and $T(1, 65537)$. Absolutely any two integers are related by T. As a relation, this is pretty silly, but this set of all possible pairs is interesting enough in its own right that it has a name and a special notation.

Definition 5.2. Given two sets X and Y, the **cartesian product** of X and Y is the set of all pairs where the first coordinate is a member of X and the second coordinate is a member of Y. We write $X \times Y$ for the cartesian product of X and Y.

This is really all there is to the cartesian product. But knowing the definition isn't the same as understanding it, so let's do a few examples.

Example 5.4. (a) Let $A = \{1, 2, 3\}$ and $B = \{a, b\}$. Write $A \times B$ in set-list notation.

$A \times B = \big\{(1, a), (1, b), (2, a), (2, b), (3, a), (3, b)\big\}$

(b) Is $(a, 2)$ a member of $A \times B$?

No. Even though the × symbol looks symmetric, it is not a commutative operator. The order matters! The pair $(a, 2)$ is a member of $B \times A$, but not $A \times B$!

(c) Is it possible for a pair to be a member of both $X \times Y$ and $Y \times X$?

Yes, but only if X and Y have some members in common. For example, if $X = \{1, 2, 3, 4\}$ $Y = \{0, 2, 4, 6\}$, then $(2, 4)$ is a member of both $X \times Y$ and $Y \times X$. (And so is $(4, 4)$!)

(d) Give an example of a member of $\mathbb{R} \times \mathbb{Z}$ that is not a member of $\mathbb{Z} \times \mathbb{R}$.

$(0.5, 2)$ will do the job. As will any pair of real numbers where the first number isn't an integer, but the second number is.

(e) Give an example of a subset of $\mathbb{Z} \times \mathbb{Z}$.

Well, we could go with the trivial subset $\varnothing$, but that doesn't tell us much. We could also try the improper subset $\mathbb{Z} \times \mathbb{Z}$, but again, that seems to be missing the point. So how about $\big\{(-1, 2), (2, -3), (-3, 2), (1, 1)\big\}$? That should do the trick.

(f) Give an example of an infinite proper subset of $\mathbb{Z} \times \mathbb{Z}$.

I can't use set-list notation anymore because I need an infinite set. One option would be to do something like $\mathbb{N} \times \mathbb{N}$ or $\mathbb{Z} \times \mathbb{N}$ or $\mathbb{Z} \times \{1, 2\}$. Make sure you know exactly what pairs are in each of these sets before moving on. Or you could use set builder notation to create an example like $\{(n, m) \mid n \geq m \wedge n, m \in \mathbb{Z}\}$.

If you're paying attention, you've probably realized that those last to example problems were really about giving examples of relations on $\mathbb{Z}$. In fact, you can think of the cartesian product $A \times B$ as the biggest possible relation from A to B, and every other relation from A to B is a subset of $A \times B$.

5.4 Properties of Relations

You may have noticed that some commonly occurring relations are very similar to each other. The $\leq$ and $\geq$ relations for real numbers are very similar to the subset $\subseteq$ and superset $\supseteq$ relations for sets. Equality (for just about anything) behaves a lot like logical equivalence does for propositional formulas. The similarities are so strong that we even write them in similar ways. We'd like to be able to use this to our advantage, so that when we're dealing with new relations like $\subseteq$, we can use what we already know about relations like $\leq$. Unfortunately, we can't just say that everything we know about $\leq$ applies to $\subseteq$; these relations aren't *exactly* the same. For example, if you pick any two different real numbers x and y, one of them has to be less than the other (either $x \leq y$ is true or $y \leq x$ is true), but if you have two different sets A and B, it's entirely possible that neither is a subset of the other (e.g, $\{1, 2\} \nsubseteq \{2, 3\}$ but $\{2, 3\} \nsubseteq \{1, 2\}$). This is one reason to talk about properties of relations. It allows us to talk about the ways in which the relations are similar to each other and makes precise our vague intuitions about their similarity.

For now, we're just going to be talking about relations on a single set and not relations from one set to another. So for the following definitions, assume that A is some set, and R is a relation on A.

Note that the definition of every one of these properties starts with some kind of universal claim like "for every..." (just like the definitions of tautologies, contradictions, logical equivalence, valid arguments, and set containment). Proving universal claims takes some work because you have to convince someone that it's true for everything. So when you want to show that a relation has a particular property, you will have to work for it, maybe even writing a proof.

On the upside, disproving universal claims is really easy. If you want to show that some relation *doesn't* have a particular property, then all you need to do is give a counterexample.[47]

Let's start with a simple property.

Definition 5.3. A relation R on a set A is **reflexive** if and only if: for every $x \in A$, $R(x, x)$.

In other words, when R is reflexive, everything in the domain is related to itself. So if your domain is the set $\{1, 2, 3\}$, then this means that a reflexive

[47] Sometimes it's hard to resist the urge to give some kind of argument to disprove a universal claim, but you should resist! I've found that most of the time, students who avoid giving counterexamples are actually trying to prove a much stronger claim. Sometimes that stronger claim isn't even true! For example, if you need to show that the relation $S = \{(a, b) \mid a = b^2\}$ on $\mathbb{R}$ is not reflexive, the easy answer is to just give a counterexample: "$(2, 2) \notin S$ because $2 \neq 2^2$." Some students start out on the right track by noting that in order for $(x, x) \in S$ to be true, $x = x^2$ must be true. But then they just say something like "this is impossible" and call it a day. But it's not impossible; it happens to be true when $x = 1$ and also when $x = 0$. The big mistake these students make is not that they forgot about 1 and 0 (anyone could make that mistake). The mistake they made was thinking that they needed to show that $x = x^2$ was impossible! They only needed to show that it was *sometimes* false. In fact, they only needed to find one situation where it was false.

relation must include the pairs $(1,1)$, $(2,2)$, and $(3,3)$. Note not *every* pair has to be of the form (x,x), as long as it has such a pair for every x in the domain.

If your relation is defined by a directed graph, reflexive relations are easy to spot because in a reflexive relation, every object in the domain will have a self-pointing arrow. (There can be other arrows too, as long as all possible self-pointing arrows are there.)

Note: The notes for the remainder of this section are not very detailed, and I apologize for that. I'll try to expand them more thoroughly when I get a chance, but I make no promises.

Example 5.5. Let $A = \{1,2,3\}$.

(a) Is $R_1 = \{(1,1),(1,2),(2,2),(3,2),(3,3)\}$ a reflexive relation on A?

Yes, it includes all of the required pairs $(1,1)$, $(2,2)$, and $(3,3)$.

(b) Is $R_2 = \{(1,1),(1,2),(2,2),(3,2)\}$ a reflexive relation on A?

No, $(3,3) \notin R_2$.

(c) Is $R_3 = \{(1,2),(1,3),(2,3),(3,2)\}$ a reflexive relation on A?

No, $\neg R_3(1,1)$.

(In this case, you could use any member of A to create a counterexample, but to prove R_3 is not reflexive, you only need to give a single example, and so you should only give a single example instead of trying to explain that everything works as a counterexample.)

(d) Is $G = \{(x,y) \mid x \geq y\}$ a reflexive relation on $\mathbb{R}$?

Yes, for any $x \in \mathbb{R}$, $x \geq x$, so $G(x,x)$.

(e) Is $D = \{(m,n) \mid n \text{ is divisible by m}\}$ on $\mathbb{Z}$ reflexive?

Yes, for any $n \in \mathbb{Z}$, n is divisible by itself, so $(n,n) \in D$.

(f) Is $T = \{(x,y) \mid x + y = 10\}$ on $\mathbb{Z}$ reflexive?

No, $2 + 2 \neq 10$, so $T(2,2)$ is false.

(g) Is $G' = \{(x,y) \mid x > y\}$ a reflexive relation on $\mathbb{R}$?

No, $\neg G'(5,5)$.

(h) Is $S = \{(s,t) \mid s \text{ ends with the same letter that } t \text{ starts with }\}$ a reflexive relation on the set of text strings?

No, S does not relate the string `"foobar"` to itself.

Let's write an actual proof of reflexivity. Let E be the relation on $\mathbb{Z}$ defined by $E = \{(n,m) \mid n + m \text{ is even }\}$.

Claim. E is reflexive.

Proof.
Choose an integer n.
$n + n = 2n$
Since n is an integer, this means $n + n$ is even, and hence $E(n, n)$. □

You may have noticed that on some of the examples above (such as R_3 and G'), you could have picked absolutely anything as a counterexample. Those relations are worse than simply "not reflexive", they're sort of the *opposite* of reflexive.

Definition 5.4. A relation R on a set A is **antireflexive** if and only if: for every $x \in A$, $\neg R(x, x)$.

Example 5.6. Let $A = \{1, 2, 3\}$.

(a) Is $R_1 = \{(1,1), (1,2), (2,2), (3,2), (3,3)\}$ an antireflexive relation on A?

No, $R_1(2, 2)$.

(In this case, you could use any member of A to create a counterexample, but to prove R_1 is not antireflexive, you only need to give a single example, and so you should only give a single example instead of trying to explain that everything works as a counterexample.)

(b) Is $R_2 = \{(1,1), (1,2), (2,2), (3,2)\}$ an antireflexive relation on A?

No, $(1, 1) \in R_2$.

(c) Is $R_3 = \{(1,2), (1,3), (2,3), (3,2)\}$ an antireflexive relation on A?

Yes, for every $x \in A$, $(x, x) \in R_3$.

(d) Is $G = \{(x, y) \mid x \geq y\}$ an antireflexive relation on $\mathbb{R}$?

No, $(17, 17) \in G$.

(e) Is $T = \{(x, y) \mid x + y = 10\}$ on $\mathbb{Z}$ antireflexive?

No, $5 + 5 = 10$, so $T(5, 5)$.

(f) Is $G' = \{(x, y) \mid x > y\}$ an antireflexive relation on $\mathbb{R}$?

Yes, for any $x \in \mathbb{R}$, x is not greater than itself, so $\neg G(x, x)$.

(g) Is $S = \{(s, t) \mid s$ ends with the same letter that t starts with $\}$ an antireflexive relation on the set of text strings?

No, S relates `"extreme"` to itself.

Definition 5.5. A relation R on a set A is **symmetric** if and only if: for every $x, y \in A$, if $R(x, y)$, then $R(y, x)$.

Symmetry is a fundamentally different kind of property from reflexivity. It doesn't quite say something about members from A. It only says something about those members that happen to be related by R. For a symmetric relation, *if* x is related to y, *then* y is related to x.

Example 5.7. Let $A = \{1, 2, 3\}$.

(a) Is $R_1 = \{(1,1), (1,2), (2,2), (2,3), (3,2)\}$ a symmetric relation on A?

No, $R_1(1, 2)$, but $\neg R_1(2, 1)$.

(b) Is $R_2 = \{(1,1), (1,2), (2,1), (2,2), (2,3), (3,2)\}$ a symmetric relation on A?

Yes, for every pair $(x, y) \in R_2$, we also have the pair $(y, x) \in R_2$.

(c) Is $R_3 = \{(1,1), (2,2), (3,3)\}$ a symmetric relation on A?

Yes, but only trivially so. The only pairs $(x, y) \in R_3$ are when $x = y$, and so the requirement to also include (y, x) in R_3 is trivially satisfied.

(d) Is $G' = \{(x, y) \mid x > y\}$ a symmetric relation on $\mathbb{R}$?

No, $(4, 1) \in G'$, but $(1, 4) \notin G'$.

(Note, we could have picked absolutely any pair $(x, y) \in G'$, and that would have worked as a counterexample, but as always, it's best to be specific.)

(e) Is $G = \{(x, y) \mid x \geq y\}$ a symmetric relation on $\mathbb{R}$?

No, $(4, 1) \in G$, but $(1, 4) \notin G$.

(In this case, we could *almost* pick any pair $(x, y) \in G$ as a counterexample. As long as $x \neq y$, it will work as a counterexample.)

(f) Is $T = \{(x, y) \mid x + y = 10\}$ on $\mathbb{Z}$ symmetric?

Yes, because addition is commutative, so any time $x + y = 10$, we also know that $y + x = 10$.

(g) Is $S = \{(s, t) \mid s$ ends with the same letter that t starts with $\}$ a symmetric relation on the set of text strings?

No, S relates `"abc"` to `"cde"`, but it does not relate `"cde"` to `"abc"`.

Let's do a proof. Let E be the relation on $\mathbb{Z}$ defined by $E = \{(n, m) \mid n - m \text{ is even }\}$. (This is not quite the same definition as the E defined a few pages back.)

Claim. E is symmetric.

Proof.
Choose integers n and m and assume that $E(n, m)$.
So $n - m$ is even.
So there exists an integer k such that $n - m = 2k$.
$m - n = -(n - m) = -2k = 2 \cdot (-k)$
Since k is an integer, so is $-k$.
So therefore $m - n$ is even, and hence $E(m, n)$. □

Some non-symmetric relations are *extremely* non-symmetric, like G'. In a relation like that, if you know that $G'(x, y)$, you automatically know that $G'(y, x)$ can*not* be true. It's impossible to have both $x > y$ and $y > x$.

The relation G isn't quite so severe, but it's close. It's *almost* impossible to have both $G(x, y)$ (meaning $x \geq y$) and $G(y, x)$ (meaning $y \geq x$). The only way to have both $x \geq y$ and $y \geq x$ is in the trivial case where $x = y$.

We capture both of these possibilities with the same definition:

Definition 5.6. A relation R on a set A is **antisymmetric** if and only if: for every $x, y \in A$, if $R(x, y)$ and $R(y, x)$, then $x = y$.

This is kind of a sneaky way of doing things, but it's especially nice when it comes to writing proofs. It's pretty easy to see how this definition applies to relations like G, but it's a little hard to see how it applies to relations like G'. It's *impossible* to have both $G'(x, y)$ and $G'(y, x)$, so the premise of the definition is *never* true. That means that the implication is trivially true. If the premise is always false, then the implication can never be false. This is a case where we're taking advantage of a triviality (any universal claim about impossible objects is trivially true) to make a definition that is not trivial.

Example 5.8. Let $A = \{1, 2, 3\}$.

(a) Is $R_1 = \{(1,1), (1,2), (2,2), (2,3), (3,2)\}$ an antisymmetric relation on A?

No, $R_1(2, 3)$ and $R_1(3, 2)$, but $2 \neq 3$.

(b) Is $R_2 = \{(1,1), (1,2), (2,1), (2,2), (2,3), (3,2)\}$ an antisymmetric relation on A?

No, $R_1(2, 3)$ and $R_1(3, 2)$, but $2 \neq 3$.

(c) Is $R_3 = \{(1,1), (2,2), (3,3)\}$ an antisymmetric relation on A?

Yes, but only trivially so. The only pairs $(x, y) \in R_3$ are when $x = y$, and so the requirement to also include (y, x) in R_3 is trivially satisfied.

(d) Is $R_4 = \{(1,2), (2,2), (3,2)\}$ an antisymmetric relation on A?

Yes, the only way you can have both $R_4(x, y)$ and $R_4(y, x)$ is when x and y are both 2 (and so $x = y$.)

(e) Is $R_5 = \{(1,2), (2,3), (3,1)\}$ an antisymmetric relation on A?

Yes, it's not possible to have both $R_5(x, y)$ and $R_5(y, x)$, and so the condition is trivially met.

(f) Is $G' = \{(x, y) \mid x > y\}$ an antisymmetric relation on $\mathbb{R}$?

Yes, it's impossible to have both $x > y$ and $y > x$.

(g) Is $G = \{(x, y) \mid x \geq y\}$ an antisymmetric relation on $\mathbb{R}$?

Yes, it's possible to have both $x \geq y$ and $y \geq x$, but only if $x = y$.

(h) Is $D = \{(n, m) \mid n \text{ is divisible by } m\}$ an antisymmetric relation on $\mathbb{Z}$?

Almost! It's really close, but $D(3, -3)$ and $D(-3, 3)$ are both true.

(i) Is $D = \{(n, m) \mid n \text{ is divisible by } m\}$ an antisymmetric relation on $\mathbb{N}$?

Yes, if n is divisible by m and m is divisible by n, then n and m are the same number. (You'll be proving this on the homework assignment.)

(j) Is $S = \{(s, t) \mid s \text{ ends with the same letter that } t \text{ starts with } \}$ an antisymmetric relation on the set of text strings?

No, S relates `"stomp"` to `"pops"` and it also relates `"pops"` to `"stomp"`.

Let's do a proof about a relation on strings. A few definitions are needed first though. If s and t are strings, we write $s + t$ to mean s concatenated with t. We say that s is a **prefix** of t if and only if there exists a string u such that $t = s + u$.

So for example, `"foo"` is a prefix of `"foobar"` because there is a string `"bar"` with ”`foobar`” = ”`foo`” + ”`bar`”. Similarly, `"foo"` is a prefix of itself because if you concatenate the empty string to `"foo"`, you get `"foo"`.

Define the relation P on the set of text strings by $P = \{(s, t) \mid s \text{ is a prefix of } t\}$.

Claim. P is antisymmetric.

Proof.
Choose strings s and t and assume that $P(s, t)$ and $P(t, s)$.
That means that s is a prefix of t and that t is a prefix of s.
So there exist strings u and v such that $t = s + u$ and $s = t + v$.
Thus $t = (t + v) + u$.
The strings v and u must be empty because otherwise the length of $t + v + u$ would be longer than the length of t.
Since u is empty, $t = s + u = s$. □

One last property. This one is a bit simpler than antisymmetry.

Definition 5.7. A relation R on a set A is **transitive** if and only if: for every $x, y, z \in A$, if $R(x, y)$ and $R(y, z)$, then $R(x, z)$.

Example 5.9. Let $A = \{1, 2, 3, 4\}$.

(a) Is $R_1 = \{(1, 1), (1, 2), (1, 3), (2, 2), (2, 3), (3, 4), (3, 3)\}$ a transitive relation on A?

No, $R_1(1, 3)$ and $R_1(3, 4)$, but $\neg R_1(1, 4)$.

(b) Is $R_2 = \{(1, 1), (1, 2), (1, 3), (1, 4), (2, 2), (2, 3), (2, 4), (3, 4), (3, 3)\}$ a transitive relation on A?

Yes, every time you have (x, y) and (y, z) in R_2, you also have (x, z).

(c) Is $R_3 = \{(1, 2), (2, 1), (2, 3), (1, 3)\}$ a transitive relation on A?

No, but this one is kind of sneaky. You have to check the situation where x and z are the same number. $(1, 2) \in R_3$ and $(2, 1) \in R_3$, but $(1, 1) \notin R_3$.

If you think about it for a few minutes, you'll realize that you fortunately don't ever have to worry about the situations where $x = y$ or when $y = z$. (But keep an eye out for when $x = z$.

(d) Is $G = \{(x, y) \mid x \geq y\}$ a transitive relation on $\mathbb{R}$?

Yes, if $x \geq y$ and $y \geq z$, then $x \geq z$.

(e) Is $D = \{(m, n) \mid n \text{ is divisible by m}\}$ on $\mathbb{Z}$ transitive?

Yes! You'll be proving this one in your homework assignment.

(f) Is $T = \{(x, y) \mid x + y = 10\}$ on $\mathbb{Z}$ transitive?

No, $T(2, 8)$ and $T(8, 2)$, but $\neg T(2, 2)$.

(g) Is $S = \{(s, t) \mid s \text{ ends with the same letter that } t \text{ starts with }\}$ a transitive relation on the set of text strings?

No. S relates `"123"` to `"345"` and also relates `"345"` to `"567"`, but it doesn't relate `"123"` to `"567"`.

Let's do a proof. Let $E = \{(n, m) \mid n - m \text{ is even }\}$ be a relation on $\mathbb{Z}$.

Claim. E is transitive.

Proof.
Choose integers l, m, and n. Assume that $E(l, m)$ and $E(m, n)$.
So $l - m$ and $m - n$ are both even.
Thus there exist integers j and k such that $l - m = 2j$ and $m - n = 2k$.
Adding those two equations together: $(l - m) + (m - n) = 2j + 2k$.
Canceling out the m's and factoring out a 2: $l - n = 2(j + k)$.
Since j and k are both integers, $j + k$ is an integer.
Therefore $l - n$ is even, and hence $E(l, n)$. □

5.5 Functions

You probably already know a fair bit about functions, both from your high school algebra classes and your programming classes. Of course, mathematical functions and programming functions are *not* the same thing, but they've got a lot n common.

Intuitive "Definition". A **function** takes an input and assigns to it an output.

This vague intuitive conception of a function isn't actually wrong. Both mathematical functions and programming functions have inputs and outputs. The big difference is that programming functions (even pure functions with no side effects) are based on instructions for *how to calculate the output.* Mathematical functions are just abstract relationships between the inputs and outputs.

If you give two seemingly different definitions for a function, but the relationships defined by the two definitions are identical, then they aren't actually different functions at all. So for example, the definitions $f(x) = (x - 10)^2$ and $g(x) = (10 - x)^2$ both define the exact same function. This is the function that maps 0 to 100, 1 to 81, 2 to 64, 3 to 49, etc. Any function that maps the same inputs to the same outputs is *the same function.* And I'm not saying that they are *equivalent* functions, but that they are literally identical: f and g are just different names for the same function, in the same way that $\{1, 2, 3\}$ is the same set as $\{2, 3, 1\}$ and $\frac{1}{2}$ is the same number as 0.5.

I used the word "relationship" to describe both relations and functions, and that's not a coincidence. When it gets down to formal definitions, a function is really just a relation with specific properties that allow you to talk about inputs and outputs. Sicne a relation is, formally speaking, just a set of pairs, a function is just a set of pairs where the first components work as inputs and the second components work as outputs.

Okay, so what kinds of properties would be necessary to be able to treat the first components as inputs and the second as outputs? To answer this, let's think about a relation that is obviously not a function and one that clearly *is* a function.

The relation $G = \{(x, y) \mid x \geq y\}$ on $\mathbb{R}$ is clearly not a function. We can't treat the pairs in G as inputs and outputs because for any given first component x, there are infinitely many possible second components y. If we try to find the "output" for the input 7, for example, we end up with too many choices. G includes pairs like $(7, 6)$, $(7, \frac{3}{2})$, $(7, -257)$, along with countless other pairs that start with 7. So there's no clear choice for which "output" is *the* output for the input 7.

On the other hand, consider a relation that is obviously a function, such as $L = \{(s, n) \mid n = \texttt{length}(s)\}$ from strings to integers. In this case, every possible string s is paired with exactly one integer. `"foobar"` is mapped to 6, because $(\texttt{"foobar"}, 6) \in L$. Similarly, `"foo"` is mapped to 3, `"bar"` is mapped to 3, `"F"` is mapped to 1, and even the empty string ε is mapped to 0.

So maybe we could say that a relation is a **function** if every member of the domain is paired with one and exactly one member of the codomain. And conceptually, this is exactly what we're going to do. Of course, the concept of "one and exactly one" is deceptively tricky. We know how to do "at least one" very easily. That's what existential claims are for. But while it's easy to *say* how many values there are with some property, it's much harder to actually *prove* how many values there are. Fortunately, there's a nifty little trick we can use when we want to prove that there is no more than one value fitting some property. We call this kind of proof a **uniqueness** proof.

The idea behind a uniqueness proof is similar to the idea behind proof by contradiction. If you're using proof by contradiction to disprove an existential claim, you assume that there is a value with the property and then use that assumption to prove a contradiction, thus showing that it's impossible to have a value with that property. In a uniqueness proof, we need to show that you can't have more than one value with a given property. So we assume that there are *two* values with the property, and then we prove that the two values are actually the same value. It's not quite a contradiction, but it has the same kind of feel to it, where you assume something that *looks* like it should be impossible (there are two solutions) and prove that it's only actually possible in the trivial case (the "two" solutions are really just one solution).[48]

Of course, this idea of a uniqueness proof isn't really a brand new proof technique. It's still just a direct proof of an if-then claim (if both y_1 and y_2 have the property, then $y_1 = y_2$), so we can build this idea into our definition of what it means for a relation to be a function.

Definition 5.8. Let R be a relation from a set A to a set B. We say that R is a **function** if and only if it fits both of the following requirements:

(uniqueness) For every $a \in A$ and every $b_1, b_2 \in B$, if $R(a, b_1)$ and $R(a, b_2)$, then $b_1 = b_2$.

(existence) For every $a \in A$, there exists a $b \in B$ such that $R(a, b)$.

The **uniqueness** requirement states that no member of the domain can be paired with two *different* members of the codomain (if it looks like you have two different members of B, then they're really the same member). In other words, no input is mapped to *more than one* output.

The **existence** requirement states that every member of the domain is paired with *at least one* member of the codomain. In other words, every input is mapped to at least one output.

Together, these two properties are the definition of what it means to be a function.

Let's look at a few examples on small, finite domains and codomains:

Example 5.10. Let $A = \{1, 2, 3, 4\}$ and $B = \{\square, \triangle, \bigcirc\}$.

(a) The relation $R_1 = \{(\square, 1), (\bigcirc, 2), (\bigcirc, 4), (\triangle, 3)\}$ from B to A is not a function because it fails the uniqueness requirement: $R_1(\bigcirc, 2)$ and $R_1(\bigcirc, 4)$, but $2 \neq 4$. It does fit the existence requirement, but honestly, nobody cares about the existence requirement if the uniqueness requirement isn't met.

(b) The relation $R_2 = \{(1, 2), (2, 3), (2, 4), (3, 3)\}$ on A fails both requirements, so it's not even close to being a function. It fails the existence requirement because there is no $y \in A$ with $R_2(4, y)$. It fails the second because $(2, 3) \in R_2$ and $(2, 4) \in R_2$, but $3 \neq 4$.

(c) The relation $R_3 = \{(1, \square), (2, \bigcirc), (4, \bigcirc), (3, \triangle)\}$ is a function from A to B. Every member of A is paired with one and exactly one member of B.

A relation like G massively fails the uniqueness requirement. Of course, to prove that it's not a function, we only need a single counterexample.

Example 5.11. The relation $G = \{(x, y) \mid x \geq y\}$ on $\mathbb{R}$ is not a function because $G(3, 2)$ and $G(3, 0)$, but $2 \neq 0$.

[48] This should remind you of the way we wrote proofs for the antisymmetry property.

In this case, we have infinitely many possible "outputs" for the "input" 3. So G isn't even close to meeting the uniqueness requirement. It does meet the existence requirement, but by itself, the existence requirement isn't super interesting.

Many non-function relations come a lot closer than G to being functions.

Example 5.12. The relation $S = \{(x, y) \mid y^2 = x\}$ on $\mathbb{R}$ is not a function because $S(9, -3)$ and $S(9, -3)$, but $3 \neq -3$.

In the above example, I proved that S was not a function by giving a counterexample for the uniqueness requirement. It turns out that S actually fails both requirements,[49] but we only needed to show one of the requirements failed to prove it was not a function.

Example 5.13. The relation $D = \{(x, y) \mid 2y = x\}$ on $\mathbb{R}$ is a function. To see that it fits the (existence) requirement, note that any real number x will be paired with the number $\frac{x}{2}$. To see that D fits the (uniqueness) requirement, note that any real number x will *only* be paired with $\frac{x}{2}$. If it looks like you have two such "outputs" (i.e., if $2y_1 = x$ and $2y_2 = x$), then they're really the same number (because $2y_1 = 2y_2$, and hence $y_1 = y_2$).

These are general explanations, not proofs, but the proofs would really be the same arguments, just with more detail. Let's do an example of such a proof right now.

Claim. Let A be a relation from $\mathbb{R}$ to $\mathbb{R}$, defined by $A = \{(x, y) \mid 3x - 2 = 2y + 1\}$. A is a function.

By now, you should have lots of experience taking the definition of a property and using it to determine the structure of the proof. Here's what the definition of the existence requirement gets us for the first part of the proof:

Proof.

Existence:

Choose $x \in \mathbb{R}$.

$\vdots$

$\cdots$, and so $A(x, \langle \textit{some real number} \rangle)$

Uniqueness:

$\vdots$

□

At the moment, I have no idea which number $\langle \textit{some real number} \rangle$ is going to be. I have to figure out what that number is, make sure it is a real number, and show that it fits the requirement $A(x, \langle \textit{some real number} \rangle)$. Note that this set-up doesn't depend on what A means; it came entirely from the structure of the definition of the (existence) requirement of being a function. Now that I've

[49] To prove that it fails the existence requirement, we could give the counterexample -9 and point out that there is no $y \in \mathbb{R}$ such that $y^2 = -9$. It's true that both $3i$ and $-3i$ are solutions to that equation, but neither of those numbers is a real number.

got the set-up and I know what my goal is, we can now unpack the definition of which relation A actually is. Our goal is to find the number that will fill the requirement $3x - 2 = 2 \cdot \langle \textit{some real number} \rangle + 1$, if there is such a number.[50]

Up until now, we've been following mostly the same strategy for finding the right formula for a value that will help us prove that an existential claim is true:

Strategy. "Forward" Strategy for Proving Existential Claims

1. Write down any true equation we can find that matches one side of the equation in our goal, using equations that we already know to be true.
2. Rewrite the other side of the equation to match the form of the equation in our goal.
3. Read the formula for our desired value from the resulting equation and verify that this value has any other required properties (such as being an integer or being non-zero).

Unfortunately, this strategy isn't super useful if you don't have any assumptions that lead to useful equations. I mean, we could start with $3x-2 = 3x-2$, but there's no obvious way to rewrite $3x-2$ in the form $2 \cdot \langle \textit{some real number} \rangle + 1$.[51] So let's try a different strategy, one where we work backwards from our goal to something more manageable. Note that when I say "work backwards", I'm only talking about our scratch work. When we actually write our proof, all our arguments need to work forwards from assumptions and obvious facts towards our conclusion.

So here's the strategy we're going to follow:

Strategy. "Backward" Strategy for Proving Existential Claims

1. Pretend (in scratch work, not the proof itself) that we already know that our desired value exists. Give that value a variable as a name and write down the goal equation using that name.
2. Solve the resulting equation for our new variable.
3. Go back to the proof and use the resulting formula to define our desired value.
4. Prove that the value meets any extra requirements.
5. Prove that the value satisfies the goal equation.

For our specific problem here, let's pretend that we already know that our $\langle \textit{some real number} \rangle$ exists and call it y. Plugging into our goal equation, we get $3x - 2 = 2y + 1$. Let's solve that for y:

[50] If we can't find such a number, then maybe this isn't actually a function after all, and we should really be looking for a counterexample.

[51] It *can* be done this way, but it's not the most natural approach.

$$
\begin{aligned}
2y + 1 &= 3x - 2 \\
2y &= 3x - 2 - 1 \\
&= 3x - 3 \\
y &= \frac{3x-3}{2}
\end{aligned}
$$

You can be as sloppy as you want here, skipping whatever steps you like; this isn't part of the proof, just some scratch work to figure out what value we want for $\langle$*some real number*$\rangle$. We don't even have to worry about making sure we find *all* the solutions, as long as we find *at least one* solution, so you can feel free to do things like take the square root of both sides without introducing a ± or anything like that.[52]

What does this scratch work tell us? Well, it tells us that the formula $\frac{3x-3}{2}$ defines a number that will satisfy the equation in our goal. So as long as that defines a number with the needed properties, we're set! In this case, the only other required property is that this number be a real number. And we know this because the real numbers are closed under multiplication, addition, subtraction, and non-zero division:

Fact. The real numbers are closed under addition, subtraction, multiplication, and division (as long as the divisor is non-zero). In other words, for any real numbers x and y, $x + y$, $x - y$, and $x \cdot y$ are real numbers. And for any real numbers x and y, if $y \neq 0$, then $\frac{x}{y}$ is a real number.

Of course, we still need to demonstrate *in our actual proof* that our newly defined real number satisfies the required equation. There is more than one way to present this. Here is one good way:

Proof.

Existence:

Choose $x \in \mathbb{R}$.

Let $y = \frac{3x-3}{2}$.

Since 2, 3, and x are real numbers, so is y.

$$
\begin{aligned}
2y + 1 &= 2 \cdot \left(\frac{3x-3}{2}\right) + 1 \\
&= (3x - 3) + 1 \\
&= 3x - 2
\end{aligned}
$$

This means that $A(x, y)$.

□

Here, the variable y is not being used for a direct proof (so *don't* write "Choose a real number y..."). It's also not being used for existential *elimination*

[52] We'll worry about showing that this is the only possible solution in the second part of the proof.

(so don't write "So there exists a real number y such that…"). This is simply a *definition* that y now means $\frac{3x-3}{2}$. Think of it as an abbreviation, if you want. You can use the word "let" as I did in this example, or you could use "define" as well. But avoid words that indicate that this is an assumption (like "assume", "choose", or "suppose"), and don't phrase it like y is some unknown quantity that exists as a consequence of an existential assumption. You don't even have to give a name to the value, if you really don't want to, although you still need to draw attention to $\frac{3x-3}{2}$ as a single value:

Proof.

Existence:

Choose $x \in \mathbb{R}$.
Consider the number $\frac{3x-3}{2}$.
This number is real because 2, 3, and x are real numbers.

$$\begin{aligned} 2 \cdot \left(\frac{3x-3}{2} \right) + 1 &= (3x-3) + 1 \\ &= 3x - 2 \end{aligned}$$

This means that $A\left(x, \frac{3x-3}{2}\right)$.

□

I wrote the part of the proof that demonstrates the required equation as a single chain, but you could also write it as a sequence of equations (one which looks a lot like the scratch work we did, only more careful and in reverse order):

Proof.

Existence:

Choose $x \in \mathbb{R}$.
Let $y = \frac{3x-3}{2}$.
Since 2, 3, and x are real numbers, so is y.

$$\begin{aligned} y &= \frac{3x-3}{2} \\ 2y &= 3x - 3 \\ 2y + 1 &= 3x - 3 + 1 \\ &= 3x - 2 \end{aligned}$$

This means that $A(x, y)$.

□

I don't care which of the above proof presentation techniques you use, as long as you are clear about what you are doing. But remember that you always have to present your proofs in a forward direction: Start with assumptions, definitions, and obvious facts, and then work from those to your goals. Never start with your goals!

Okay, so now that we've shown several ways for presenting the proof of the existence requirement (that every "input" has at least one "output"), let's move onto the uniqueness requirement (that you can't have two "outputs" for the same "input"). Following the structure of the definition gives us this structure:

Proof.

Uniqueness:

Choose real numbers x, y_1, and y_2 and assume that $A(x, y_1)$ and $A(x, y_2)$.

$\vdots$

$\cdots$, and so $y_1 = y_2$.

□

Again, this structure comes entirely from the definition of the uniqueness requirement and what we know about proving universal claims. You can get this far without even thinking about what A means. Of course, we do know what A means, so there's an obvious next step:

Proof.

Uniqueness:

Choose real numbers x, y_1, and y_2 and assume that $A(x, y_1)$ and $A(x, y_2)$.
So we know $3x - 2 = 2y_1 + 1$ and $3x - 2 = 2y_2 + 1$.

$\vdots$

$\cdots$, and so $y_1 = y_2$.

□

Now it's just an algebra problem. We have two equations involving y_1 and y_2 and x and we need to combine them in a way that will hopefully result in being able to prove $y_1 = y_2$. We could do substitution, add the equations, subtract them, multiply them, etc. Since we know that our final goal $y_1 = y_2$ doesn't have an x in it, a method that eliminates x is probably a good idea. We could solve one equation for x and then plug it into the other equation, and that would work just fine. We could even subtract the two equations from each other, and that would work great too. But since both equations are currently in the form $3x - 2 = \cdots$, the easiest thing to do is probably just to set the opposite sides equal. Since both $2y_1 + 1$ and $2y_2 + 1$ are equal to $3x - 2$, they must be equal to each other.

Proof.

Existence:

Choose $x \in \mathbb{R}$.
Let $y = \frac{3x-3}{2}$.
Since 2, 3, and x are real numbers, so is y.

$$\begin{aligned} 2y + 1 &= 2 \cdot \left(\frac{3x - 3}{2} \right) + 1 \\ &= (3x - 3) + 1 \\ &= 3x - 2 \end{aligned}$$

This means that $A(x, y)$.

Uniqueness:

Choose real numbers x, y_1, and y_2 and assume that $A(x, y_1)$ and $A(x, y_2)$.
So we know $3x - 2 = 2y_1 + 1$ and $3x - 2 = 2y_2 + 1$.

$$\begin{aligned} 2y_1 + 1 &= 2y_2 + 1 \\ 2y_1 &= 2y_2 \\ y_1 &= y_2 \end{aligned}$$

□

For simple linear equations like these, the end result is a proof that seems obvious to the point of stupidity, but you need to be careful to make sure each step is valid. In that first step, we subtracted 1 from both sides, which is perfectly legal. In the second, we divided both sides by 2, which is also totally fine. You need to be careful about things because when something fails, it's often a very subtle thing that trips you up. For example, make sure you aren't dividing both sides by a variable (or expression) that might sometimes be zero. Similarly, be careful when taking the square root of both sides of an equation.

Remember that non-function relation $S = \{(x, y) \mid y^2 = x\}$ on $\mathbb{R}$? If you thought that this might be a function, you could get all the way through proving the existence requirement and part of the way through the uniqueness requirement:

Proof.

Uniqueness:

Choose real numbers x, y_1, and y_2 and assume that $S(x, y_1)$ and $S(x, y_2)$. So we know $y_1^2 = x$ and $y_2^2 = x$.

$$\begin{aligned} y_1^2 &= y_2^2 \\ &\vdots \end{aligned}$$

□

And if you tried to take the square root of both sides, you might incorrectly end up with $y_1 = y_2$. But S is not a function, so that must mean you made a mistake! If you're careful about it, you'll realize that taking the square root of both sides results in $|y_1| = |y_2|$ or maybe $y_1 = \pm y_2$, which doesn't prove that $y_1 = y_2$. So be careful!

5.5.1 Function Notation

If you know that a relation is a function, then there are some special notations you can make use of. We often write $f : A \to B$, which means "f is a function from A to B". This identifies the domain A and the codomain B, *and* it states that f is a function, not just any old relation.

With relations, when we have a pair (x, y) in the relation, we might say that the relation "relates x to y". If we know that the relation is a function, we can also say that the function "**maps** x to y", or that it "**assigns** the output y to x", or that it "**sends** x to y."

There's also a notation (called **function notation**) you can use to describe the output that goes with some specific input. If you have a function $f : A \to B$

and you have a member a of the domain, then we write $f(a)$ to mean "the output that a is mapped to." Note that this only makes sense if f is a function. If there are multiple possible values that a is related to, then which one is "the" output?

Example 5.14. Let $A = \{1, 2, 3, 4\}$ and $B = \{\square, \triangle, \bigcirc\}$. Let $R_3 : A \to B$ be defined by $R_3 = \{(1, \square), (2, \bigcirc), (4, \bigcirc), (3, \triangle)\}$. Define R_4 from B to A by $R_4 = \{(\square, 2), (\triangle, 4), (\bigcirc, 1)\}$.

(a) $R_3(3) =?$

Since $(3, \triangle) \in R_3$, we can say that $R_3(3) = \triangle$.

(b) Is there an x such that $R_3(x) = \bigcirc$?

Yes. There are actually two! $R_3(2) = \bigcirc$ and $R_3(4) = \bigcirc$.

(c) $R_4(\triangle) =?$

$R_4(\triangle) = 4$

(d) Is there an x such that $R_4(x) = 3$?

Nope!

(e) $\big(R_4(\square) + 2\big)^2 =?$

$\big(R_4(\square) + 2\big)^2 = (2 + 2)^2 = 4^2 = 16$

(f) $R_4\big(R_3(2)\big) =?$

$R_4\big(R_3(2)\big) = R_4(\bigcirc) = 1$

In the process of proving that a relation is a function, you'll often find yourself solving for the output variable. (The output variable is often y, but not always.) In fact, when I encounter a relation defined using an equation and I want to know if it's a function or not, solving for the "output" variable is usually the very first thing I do. If you can rewrite the equation in the form $y =$ something involving x, then you've already done most[53] of the work needed for showing that the relation is a function.

Example 5.15. If the property that defines the relation is already in the form "$\langle$*output variable*$\rangle =$ some formula involving $\langle$*input variable*$\rangle$", then it's pretty easy to tell whether the relation is a function or not.

(a) $F = \{(x, y) \mid y = \frac{x^2+1}{2}\}$ is a function on $\mathbb{R}$. If you choose a real number x, then $\frac{x^2+1}{2}$ is a specific well-defined real number.

[53] You do need to make sure that the expression involving x is properly defined for all possible values of x and that the resulting value is in the codomain, and you do need to make sure that this is the *only* possible solution, but those are usually pretty easy.

(b) $N = \{(x, y) \mid y = \frac{x^2+1}{2}\}$ is not a function on $\mathbb{Z}$. It's got the right format, but when $x = 2$, $\frac{x^2+1}{2} = \frac{4+1}{2} = \frac{5}{2}$, which is not an integer. So N fails the existence requirement: there is no integer y with $N(2, y)$.

(c) The relation $L = \{(s, n) \mid n = \text{ the length of } s\}$ from the set of all strings to $\mathbb{Z}$ is a function. Every string has a length, and that length is always in integer.

It's so easy to think about functions when they are given in this format that we often combine this strategy with function notation when we define a function. In fact, this is probably the notation you are most familiar with for defining functions. If you can write the output as a formula involving an input variable, then you can use function notation like this: "$f(x) = \text{ some formla involving } x$". This is really shorthand for "For every input x in the domain, the output for x is 'some formula involving x'." Or in other words, $f = \{(x, y) \mid y = \text{ some formula involving } x\}$.

So for example, I could redefine the example F above by just writing $F(x) = \frac{x^2+1}{2}$, and you could redefine L by $L(s) = \text{ the length of } s$.

5.5.2 Partial Functions

If you define a "function" using function notation as in these examples, it's not automatically guaranteed to truly be a function. You do need to check to make sure that the formula used is well-defined on the entire domain, and you need to make sure that the resulting values are actually in the codomain.

Example 5.16. Here are a few non-function relations defined using function notation.

(a) $s : \mathbb{R} \to \mathbb{R}$ defined by $s(x) = \sqrt{x}$.

This isn't a function because $\sqrt{-4}$ is not in the codomain $\mathbb{R}$. It's common practice to still use this notation and to just say that $s(-4)$ is **undefined** if the obvious value for $s(-4)$ isn't in the codomain.

(b) Let I be defined on $\mathbb{R}$ by $I(x) = \frac{1}{x}$.

This is the familiar additive inverse "function". It's really, really close to a function, but $I(0)$ is undefined because $\frac{1}{0}$ is undefined. So technically speaking, this isn't a function at all.

(c) Let f be defined on the set of all text strings as $f(s) = \text{ the first character in } s$.

This one looks like it is well-defined, but it's not quite a function because $f(\varepsilon)$ is undefined. The empty string doesn't have a "first character".

You may have noticed that all of the above not-quite functions meet the uniqueness requirement for being a function (no input has more than one output), but they don't quite meet the existence requirement (every input has an output). This isn't nearly as bad as failing the uniqueness requirement, so it still sort-of makes sense to use the language of functions when talking about them. We can still talk about “inputs” and “outputs” as long as we keep in mind the fact that some inputs don’t get any outputs. We can still use function notation as long as we understand that sometimes $f(x)$ just isn’t going to be defined.

These almost functions that pass the uniqueness requirement but not necessarily the first are called **partial functions**. More formally:

Definition 5.9. A relation R from A to B is a **partial function** if and only if: for every $a \in A$ and every $b_1, b_2 \in B$, if $R(a, b_1)$ and $R(a, b_2)$, then $b_1 = b_2$.[54]

If you have a function and you want to emphasize that it really is a function and not just a partial function, then you can call it a **total** function. Technically “total function” and “function” mean the same thing, but it’s useful to have the word “total” so you can emphasize that you really do mean a function.

Partial functions are actually fairly common in computer science because every programming function[55] defines a partial function, but not necessarily a total function. Why? Well some inputs will cause the function to get stuck in an infinite loop or they might cause an error to be raised. In those cases, there won’t be an output, so it can’t be represented formally by a total function. But it can be represented by a partial function. So even though partial functions aren’t used often in mathematic, they are extremely common in theoretical computer science.

Note that technically speaking, every function is also a partial function. Since a function fits requirement 2, it counts as a partial function. In ordinary English, “partial” usually means “not total”, but to a theoretician, “partial” usually means “not *necessarily* total”. You’ll see the same connection between partial orderings (where some pairs of things might not be orderable) and total orderings (where any two pairs of things will have a big one and a small one). Every total ordering is a partial ordering too, but not the other way around. It’s a little confusing, but you’ll get used to it. I find it helpful to remember the following fact:

Fact. Every relation falls into one of these three categories:

- not a partial function: fails the (uniqueness) requirement, may or may not meet the (existence) requirement (Obviously not a total function either.)
- a partial function but not a total function: meets the (uniqueness) requirement, but not the (existence) requirement
- a (total) function: meets both requirements (Technically also a partial function.)

[54] Yes, this is exactly the same as requirement 2 for being a function.

[55] At least the ones that don’t use side effects to get input from somewhere else.

5.5.3 Properties of Functions

I apologize for the lack of a detailed set of notes for this section. I will include the important definitions and a series of examples but probably not any detailed explanations.

Definition 5.10. A function $f : A \to B$ is **one-to-one** (or a **injection**) if and only if: for every $a_1 \in A$ and every $a_2 \in A$, if $f(a_1) = f(a_2)$, then $a_1 = a_2$.

This basically means that a **one-to-one** function cannot send two different inputs to the same output.

Definition 5.11. A function $f : A \to B$ is **onto** (or a **surjection**) if and only if: for every $b \in B$, there exists an $a \in A$ such that $f(a) = b$.

This basically means that for an **onto** function, every member of the codomain appears as an output for at least one member of the domain.

Example 5.17. Let f be a function on the real numbers defined by $f(x) = x^2 - 2$.

f is not one-to-one because $f(3) = 3^2 - 2 = 7 = (-3)^2 - 2 = f(-3)$, but $3 \neq -3$.

f is not onto because there is no $x \in \mathbb{R}$ with $f(x) = -10$. (There is an *imaginary* number that solves the equation $x^2 - 2 = -10$, but the domain is only the *real* numbers.)

Example 5.18. $g : \mathbb{R} \to \mathbb{R}$ is defined by $g(x) = 7x - 2$.

If you have a specific real number as your output y, you can always find an input x by solving $y = 7x - 2$ for x. Since this is a simple linear equation, there will always be a solution in the real numbers (onto), and there will always be a *unique* solution (one-to-one). (This is just an explanation, not a full proof. See below for proofs.)

Claim. g is one-to-one.

Proof.
Choose $x_1, x_2 \in \mathbb{R}$ and assume $g(x_1) = g(x_2)$.
So $7x_1 - 2 = 7x_2 - 2$.
$7x_1 = 7x_2$
$x_1 = x_2$ □

Claim. g is onto.

Proof.
Choose $y \in \mathbb{R}$.
Let $x = \frac{y+2}{7}$.
Since y, 2, and 7 are all real numbers, so is x.

$$\begin{aligned} x &= \frac{y+2}{7} \\ 7x &= y + 2 \\ 7x - 2 &= y \end{aligned}$$

Hence $g(x) = y$. □

A function that is both one-to-one and onto is called a **bijection**. Bijections are said to be **invertible**, because if you "invert" a bijection by swapping the inputs and outputs, you will get another function, called the **inverse function**.

Example 5.19. Define $h : \mathbb{R} \to \mathbb{Z}$ by $h(x) = \lfloor x \rfloor$, where $\lfloor x \rfloor$ is x rounded down to an integer (the **floor** function).

h is clearly not one-to-one. It's about as far away from one-to-one as you can get. But to prove it's not one-to-one, we only need one counterexample: $h(3.7) = 3 = h(3.127)$, but $3.7 \neq 3.127$.

Every single integer occurs as an output for h, so h is onto (proof below).

Claim. h is onto.

Proof.
Choose $y \in \mathbb{Z}$.
Let $x = y + 0.1$.
Since y and 0.1 are both real numbers, so is x.

$$\begin{aligned} h(x) &= h(y + 0.1) \\ &= \lfloor y + 0.1 \rfloor \\ &= y && \text{(because } y \in \mathbb{Z}\text{)} \end{aligned}$$

Hence $h(x) = y$. □

Often, when I ask students for extreme examples of functions, I get suggestions like $z(x) = x \cdot 0$, with some kind of trick to force the output to always be the same number. A function that always outputs the same value is a perfectly reasonable function (if a bit silly), but there's no reason to give a fancy "rule" that just happens to output 0 all the time. If you want a function that always outputs the same value, just say that the function always outputs zero: $z(x) = 0$. Such a function is called a **constant** function.

Example 5.20. Let $z(x) = 0$ define a function from $\mathbb{R}$ to $\mathbb{Z}$. This is about as far away from a

z is clearly not one-to-one because $z(5) = 0 = z(-2.5)$, but $5 \neq -2.5$.

z is clearly not onto because there is no $x \in \mathbb{R}$ with $z(x) = 17$.

Let's do some examples that aren't all about numbers. For the following examples, let Str be the set of all strings. (If we need to be specific, let's say that it's the set of all *Unicode* strings, but we probably won't need that detail.)

Example 5.21. Define a function $a_1 : \text{Str} \to \mathbb{Z}$ by:

$$a_1(s) = \text{the number of lowercase } \texttt{a}\text{'s in } s$$

a_1 is not one-to-one because $a_1(\texttt{"banana"}) = 3 = a_1(\texttt{"aaa"})$ and $\texttt{"banana"} \neq \texttt{"aaa"}$.

This function is not onto because there is no string s that has $a_1(s) = -3$ (no string can have a negative number of `a`'s in it.

Whether or not a function is onto often hinges upon the specific choice of the codomain. A small change to the codomain can turn a non-onto function into an onto function:

Example 5.22. Define a function $a_2 : \text{Str} \to \mathbb{N}$ by:

$$a_2(s) = \text{the number of lowercase } \texttt{a}\text{'s in } s$$

The only difference between a_1 and a_2 is the fact that the codomain of a_1 was $\mathbb{Z}$ and the codomain of a_2 is $\mathbb{N}$. This doesn't change the fact that a_2 is still not one-to-one, but it does turn the function into an onto function.

Claim. a_2 is onto.

Proof.
Choose $n \in \mathbb{N}$.
Let $s = n$ copies of `a` concatenated together.
Clearly s is a string, and since it has n `a`'s in it, $a_2(s) = n$. □

Example 5.23. Let $x : \text{Str} \to \text{Str}$ be defined by $x(s) = s + \texttt{"!"}$ (where $+$ is the concatenation operator).

x can never output the string `"foobar"` because it can only output strings that end with an exclamation point. So x is clearly not onto. But it *is* one-to-one.

Claim. x is one-to-one.

Proof.
Choose $s_1, s_2 \in \text{Str}$ and assume $x(s_1) = x(s_2)$.
So $s_1 + \texttt{"!"} = s_2 + \texttt{"!"}$.
Removing the last character from both sides of this equation results in $s_1 = s_2$.
□

If you want the inputs to your function to be numbers, then your domain needs to be a *set* of numbers (like $\mathbb{Z}$ or $\mathbb{R}$). If the inputs to your function are going to be strings, then your domain needs to be a *set* of strings (this is why I defined the set Str earlier). If you want the inputs to your function to be sets, then your domain will need to be a *set* whose members are also sets. So if we set the domain to be something like $\mathcal{P}(\mathbb{Z})$ (the set of all sets of numbers), then that means that the inputs to our function will be sets of numbers.

Defining such a function can be a bit tricky, because the function needs to work on both finite and infinite sets. So I could *try* to define a function $c : \mathcal{P}(\mathbb{Z}) \to \mathbb{N}$ by writing $c(X) = |X|$. This takes a set and returns the cardinality of that set (the number of members). And this will work on many sets: $c\big(\{-2, 0, 2\}\big) = |\{-2, 0, 2\}| = 3$ and $c(\emptyset) = |\emptyset| = 0$, but it fails on infinite sets because the cardinality of an infinite set is not a natural number. If I define c like this, then I'm really only defining a *partial* function.

If I wanted to turn this into a total function, there are a couple of ways to fix the problem. I could modify the definition by putting in a special case for infinite sets:

Define $c_1 : \mathcal{P}(\mathbb{Z}) \to \mathbb{Z}$ by the following:

$$c_1(X) = \begin{cases} |X| & X \text{ is finite} \\ -1 & X \text{ is infinite} \end{cases}$$

Or you could shrink the domain down to just finite sets.

Example 5.24. Let $\mathcal{P}^{\text{fin}}(\mathbb{Z})$ be the set of all *finite* subsets of $\mathbb{Z}$. Define $c : \mathcal{P}^{\text{fin}}\,\mathbb{Z} \to \mathbb{N}$ by $c(X) = |X|$.

The function c is not one-to-one because $c\big(\{-2, 0, 2\}\big) = 3 = c\big(\{1, 2, 3\}\big)$.

Claim. c is onto.

Proof.
Choose $n \in \mathbb{N}$.
Let $X = \{k \mid k > 0 \wedge k \leq n\}$. (This is just a way of writing $\{1, 2, \ldots, n\}$ without having to use the dot-dot-dot notation.)
All the members of X are integers, and there are only finitely many of them, so $X \in \mathcal{P}^{\text{fin}}(\mathbb{Z})$.
And since X clearly has n members, $c(X) = |X| = |\{k \mid k > 0 \wedge k \leq n\}| = n$.
□

Example 5.25. Define $d : \mathcal{P}(\mathbb{Z}) \to \mathcal{P}(\mathbb{Z})$ by $d(X) = \{2x \mid x \in X\}$. This looks like a complicated definition, but it's really pretty straightforward. This function takes a set X and returns a the same set but with all the members doubled. So for example, $d\big(\{2, 3, 5\}\big) = \big\{2x \mid x \in \{2, 3, 5\}\big\} = \{2x \mid x = 2 \vee x = 3 \vee x = 5\} = \{4, 6, 10\}$.

This definition even works for infinite sets. So $d(\mathbb{N}) = \{2x \mid x \in \mathbb{N}\}$, which is really just the set of all non-negative even numbers. Or if your input is the set of all multiples of 3:

$$\begin{aligned} d\big(\{3n \mid n \in \mathbb{Z}\}\big) &= \big\{2x \mid x \in \{3n \mid n \in \mathbb{Z}\}\big\} \\ &= \{2x \mid x = 3n \text{ for some } n \in \mathbb{Z}\} \\ &= \{2 \cdot (3n) \mid n \in \mathbb{Z}\} \\ &= \{6n \mid n \in \mathbb{Z}\} \end{aligned}$$

...then the output is the set of all multiples of 6.

This function is definitely not onto. You may have noticed that no matter what the input set is, the output set will only ever have even numbers as members. So you can't have a set like $\{1, 2, 3\}$ as output. More formally:

Claim. There is no set $X \in \mathcal{P}(\mathbb{Z})$ with $d(X) = \{1, 2, 3\}$.

Proof.
Suppose towards a contradiction that there is a set $X \in \mathcal{P}(\mathbb{Z})$ with $d(X) =$

$\{1, 2, 3\}$.
That means that $\{1, 2, 3\} = \{2x \mid x \in X\}$, and hence $1 \in \{2x \mid x \in X\}$.
So there exists an $x \in X$ with $1 = 2x$.
Hence $x = \frac{1}{2}$.
But this is impossible, because $\frac{1}{2}$ is not an integer and every member of X needs to be an integer (since $X \in \mathcal{P}(\mathbb{Z})$). □

5.5.4 When is a "proof" really needed?

Question. When do you need to actually write a proof to show that your counterexample works?

Unfortunately, there's no absolute answer to this question. It depends on context. There are some situations where you absolutely don't need to write a proof. If you're giving a counterexample to a purely universal claim (such as a showing a function is not one-to-one or that a relation is not transitive), then even in a situation where absolute proof is required, that just means giving the counterexample and showing that it meets the requirements.

Why is this? Well let's take a look at the definition for a function $f : A \to B$ being one-to-one and put it into the language[56] of first order logic:

$$\forall a_1, a_2 \in A\big(f(a_1) = f(a_2) \to a_1 = a_2\big)$$

And we want to prove that this is *false*. So let's slap a $\neg$ on the beginning and see if we can rewrite this in a more useful way.

$$\begin{aligned}\neg\forall a_1, a_2 \in A\big(f(a_1) = f(a_2) \to a_1 = a_2\big) &= \exists a_1, a_2 \in A\,\neg\big(f(a_1) = f(a_2) \to a_1 = a_2\big)\\ &= \exists a_1, a_2 \in A\big(f(a_1) = f(a_2) \wedge \neg(a_1 = a_2)\big)\\ &= \exists a_1, a_2 \in A\big(f(a_1) = f(a_2) \wedge a_1 \neq a_2\big)\end{aligned}$$

In other words, we just need to give examples of a_1 and a_2 and check off the requirements: they need to be members of the domain A, their outputs need to be the same, and they need to be different from each other.

But for a universal-existential claim (such as showing that a function is not onto or that a relation doesn't meet the (existence) requirement for being a function), the counterexample has a harder job to fill. You still need to give the counterexample, but proving that your counterexample works might be harder.

Take a look at the definition for a function $f : A \to B$ being onto in terms of first-order-logic:

$$\forall b \in B\Big(\exists a \in A\big(f(a) = b\big)\Big)$$

[56] I'm abusing notation here quite a bit, but we're not looking for a fully formal formula of formal logic here; we just want to write this in such a way that we can use what we know about first-order logic.

And if we negate this:

$$\begin{aligned}\neg\,\forall b \in B\Big(\exists a \in A\big(f(a) = b\big)\Big) &= \exists b \in B\,\neg\Big(\exists a \in A\big(f(a) = b\big)\Big)\\ &= \exists b \in B\Big(\forall a \in A\,\neg\big(f(a) = b\big)\Big)\\ &= \exists b \in B\Big(\forall a \in A\big(f(a) \neq b\big)\Big)\end{aligned}$$

So after showing that b exists and that it is a member of the codomain B, we have to prove a *universal* claim: that *no* member of the domain A has b as its output. Or (if you look at the second formula in that chain), we could prove that *there does not exist* such an a in the domain. Either way, we have to prove a universal claim, and that requires more work.

Sometimes that extra work is pretty easy, if it hinges upon a very simple fact (such as the fact that if you square a real number, you never get a negative number), and in cases like that, there's definitely no need to write out a full-fledged "proof". And sometimes it's not obvious at all: check out your homework bonus problem about the function $s(X) = \{n+m \mid n \in X \wedge m \in X\}$ on $\mathcal{P}(\mathbb{Z})$. For problems like that, you absolutely need to give a proof.

There will always be a gray area in the middle where you have to make a judgment about your audience and what will seem obvious to them and what needs a more detailed explanation. You may even have to balance this against space concerns, if your proof is intended for publication. For this class, where space is not at a premium, you should alway err on the side of providing too much detail. More detail is better than not enough detail.

Mathematical Induction

For this section, the universal set is assumed to be $\mathbb{N} = \{0, 1, 2, 3, \ldots\}$, the set of natural numbers. We're going to be talking about things you can prove (informally) for all natural numbers, so all of our examples will be universal claims. Make sure you remember all your basic algebra for how to manipulate equations and inequalities that deal with polynomials, exponents, etc.

Question. If n is a natural number, then which is bigger, $3n$ or 2^n?

Well let's look at it for different values of n:

n	$3n$	2^n
0	0	1
1	3	2
2	6	4
3	9	8
4	12	16
5	15	32
6	18	64

Given the above table, you can certainly say that when n is 0, 4,5, or 6, 2^n is bigger and that when n is 1, 2, or 3, $3n$ is smaller. What do you think about 7, 8, 9, etc.? As n gets bigger, both 2^n and $3n$ get bigger, but each time, it seems like 2^n increases more than $3n$ does. Just to be sure that this is true, let's look at exactly how big these increases are. When we go from $n = 4$ (row 4) to 5 (row 5), $3n$ increases by 3 (from 12 to 15) and 2^n doubles, increasing by 16 (from 16 to 32). When we go from $n = 5$ to 6, $3n$ increases by 3 (from 15 to 18) and 2^n doubles, increasing by 32 (from 32 to 64). This description (that $3n$ increases by 3 and 2^n doubles) should hold no matter which value of n we start with. If you don't believe me, give me some row number (call it k, but make sure it's at least 4) and I'll show you how to prove for that row.

Row k is going to have the number k in the first column (representing that $n = k$ for this row). What's it going to have in the second column? Three times whatever k is (in other words, $3k$). What's in the third column? That would be whatever number 2^k turns out to be. But what we're really interested in is what happens when we go to the next row, which should be row $k+1$. In the row for $n = k+1$, what's in the second column? Three times $k+1$, of course, which we write $3(k + 1)$. Now we're claiming that this $(3(k + 1))$ is just three more than what we had in the previous row ($3k$), and since $3(k+1) = 3k+3$, it's clear that this is true. What's in the third column? 2^{k+1}, that's what. We're also claiming that this (2^{k+1}) is twice what was in the previous row (2^k). This isn't as obvious, but we can show it using simple rules of exponents: $2^{k+1} = 2^k \cdot 2^1 = 2^k \cdot 2$. Of course, what really matters to us was that doubling 2^k increases it by more than adding 3 does, but as long as 2^k is bigger than 3 (which it is), this is true.

This seems like enough to prove that for any $n \geq 4$, 2^n is bigger than $3n$. I'm betting that most of you are comfortable with this line of reasoning (at least the first part), but let's examine it more carefully anyway.

We're trying to argue that for any $n \geq 4$, some particular fact is true. In this case, we're arguing that $2^n > 3n$. We can see that it's true when $n = 4$ just by calculating. We can also describe how the two values change when we increase n by 1, regardless of what value n actually is. In other words, when it's true for some value k, it must also be true for $k + 1$. These are the two key steps to **mathematical induction**:

- Show that it's true for the smallest value, and then
- Prove that *if* it is true for some value, it is also true for the next value.

The first part is called the **base step** (or **base case**). The second part, is called the **induction step**. Notice that in the base step, we're simply proving that our claim is true for a particular value, without any hypotheses other than basic algebra. In the induction step, we're proving a conditional ("if. . . then") claim, which means that the second step will always start with a hypothesis, and it will always have the same form: we assume that our claim is true for some particular value (we'll usually call it k). This assumption is called the **induction hypothesis**. (In the above example, the induction hypothesis was $2^k > 3k$.) Notice that in this step, we won't be *proving* that it's true for

k. We're *assuming* that it is true, so we can use it to prove something else. What we're trying to prove is that the claim is true for the *next* value (that is, for $k+1$). So after you make this assumption, the strategy is to look at the important numbers for the $k+1$ step (in our example, that's what $3(k+1)$ and 2^{k+1} were). We don't know anything about how they're connected, but we have a hope for how they should be connected (in this case, we're hoping to prove $2^{k+1} > 3(k+1)$. So instead, we try to connect those numbers to the ones from the k step (for us, $3k$ and 2^k). (In our example, that connection was that 2^{k+1} is twice 2^k and that $3(k+1)$ is 3 more than $3k$.) Once you have that connection, then you can actually use the assumption you made about the k numbers (that $2^k > 3k$) to say something about how the $k+1$ numbers are connected. (Since we already assumed that $2^k > 3k$, then doubling the bigger number (2^k) to get 2^{k+1} will result in a bigger number than adding 3 to the smaller number ($3k$) to get $3(k+1)$, so $2^{k+1} > 3(k+1)$.)

Why is this enough? In a sense, every induction proof is really a set of instructions for how to build a proof for any value bigger than your base case. The first part tells you how to get started, and the second part tells you how to go from one step to the next. Once you know where the foot of the stairs is and how to climb from one step to the next, you can climb to wherever you want. In math, of course, we're only interested in showing that you can get there, so we don't actually climb all the steps. We just show you how to get to the first one and then show you how to go from step k to step $k+1$, no matter what k is.

Using these ideas, let's rebuild our argument for why $2^n > 3n$ for $n \geq 4$ in a format that leaves out all this extra explanation I was giving to show you why this works. I expect you to write your proofs in a similar way to this.

Example 5.26. Prove that for every $n \geq 4$, $2^n > 3n$.

Proof. (induction on n)
(Base step, $n = 4$):
$2^4 = 16 > 6 = 3 \cdot 4$

(Induction step):
Assume that $2^k > 3k$ for some $k \geq 4$.
(Goal: $2^{k+1} > 3(k+1)$)

$$\begin{aligned} 2^{k+1} &= 2^k \cdot 2^1 \\ &= 2^k \cdot 2 \\ &> 3k \cdot 2 && \text{(by the induction hypothesis)} \\ &= 3k + 3k \end{aligned}$$

Since $k \geq 4$, we know that $3k \geq 12$, and more importantly, $3k > 3$. so:

$$3k + 3k > 3k + 3 = 3(k + 1)$$

Therefore $2^{k+1} > 3(k + 1)$. □

Let's do some other examples.

Example 5.27. Prove that for any $n \geq 1$, $n < 2^n$.

Proof. (induction on n)
(Base Step, $n = 1$):
$1 < 2 = 2^n$

(Induction Step):
Assume $k < 2^k$ for some $k \geq 1$.
(Goal: $2^{k+1} > k + 1$)

$$\begin{aligned} 2^{k+1} &= 2^k \cdot 2^1 \\ &= 2^k \cdot 2 \\ &> k \cdot 2 && \text{(by the induction hypothesis)} \\ &= k + k \end{aligned}$$

Since $k \geq 1$, we know $k + k \geq k + 1$, so this gives us:

$$2^{k+1} > k + k \geq k + 1$$

Therefore $2^{k+1} > k + 1$. □

Example 5.28. Prove that for any $n > 0$, $1 + 2 + 3 + \cdots + n = \frac{n(n+1)}{2}$.

Proof. (induction on n)
(base step, $n = 1$):
$1 = \frac{2}{2} = \frac{1(1+1)}{2}$

(induction step):
Suppose that $1 + 2 + 3 + \cdots + k = \frac{k(k+1)}{2}$ for some $k > 0$.
(will prove: $1 + 2 + 3 + \cdots + (k + 1) = \frac{(k+1)(k+1+1)}{2}$)

$$\begin{aligned}1+2+3+\cdots+(k+1) &= 1+2+3+\cdots+k+(k+1)\\ &= \frac{k(k+1)}{2}+(k+1) && \text{(by IH)}\\ &= \frac{k(k+1)}{2}+\frac{2(k+1)}{2}\\ &= \frac{k(k+1)+2(k+1)}{2}\\ &= \frac{(k+1)(k+2)}{2} && \text{(factoring out } (k+1))\\ &== \frac{(k+1)(k+1+1)}{2}\end{aligned}$$

Therefore $1+2+3+\cdots+k+(k+1) = \frac{(k+1)(k+1+1)}{2}$. □

5.6 Factorials

Simply put, the **factorial** of a natural number n (which is written $n!$) is the product of all the positive integers from 1 up to n. In other words, $n! = 1 \cdot 2 \cdot 3 \cdot \cdots \cdot n$.

For the same reason that $2^0 = 1$, we define $0! = 1$. It may or may not be obvious to you from the definition that this is the only sensible meaning for $0!$, but it is.

Example 5.29. Compute 4!.
$4! = 1 \cdot 2 \cdot 3 \cdot 4 = 2 \cdot 3 \cdot 4 = 6 \cdot 4 = 24$

Example 5.30. Compute 5!.
$5! = 1 \cdot 2 \cdot 3 \cdot 4 = 2 \cdot 3 \cdot 4 \cdot 5$. Note that computing 5! is the same as computing 4! and then multiplying by 5. So $5! = 4! \cdot 5 = 24 \cdot 5 = 120$.

This is the key to using factorials in induction proofs. No matter what number you start with (let's call it n), its factorial ($n!$) is always going to be that number times the factorial of the number one smaller than it ($(n-1)!$). As a single equation: $n! = (n-1)! \cdot n$. Of course, when we're doing an induction proof that involves $n!$, we'll usually have assumed something about $k!$ for some value of k and we'll be trying to prove something about $(k+1)!$. So a good first step is to try and write $(k+1)!$ in terms of $k!$. This works the same way: $(k+1)! = k! \cdot (k+1)$.

Example 5.31. Prove that for any $n > 3$, $2^n < n!$.

The structure of this proof follows the same structure that all induction proofs follow:

- Base Step: plug in the smallest value for n and show that the claim is true. (Usually this is just a simple computation.)

- Induction Step:
  - Assume that the claim is true for some value (call it k). (This is an assumption, a fact that we can use later in the induction step.)
  - Look at the left-hand side of the formula for the value $k+1$. Write this number in a way that uses the left-hand side for the value k. (Usually, this involves basic algebra.) [57]
  - Use the induction hypothesis (which is true for k) to connect this number (which we wrote using the left-hand side for k), to a number that is written which is written using the right-hand side for k.
  - Use more algebra to connect this number that uses the right-hand side for k to the right-hand side for $k+1$.
  - Profit!

Proof. (Induct on n)

(Base Step $n = 4$):
$2^4 = 16 < 24 = 4!$

(Induction Step):
Assume that for some $k > 3$, $2^k < k!$.
(I will show: $2^{k+1} < (k+1)!$.)

$$\begin{aligned} 2^{k+1} &= 2^k \cdot 2 \\ &< k! \cdot 2 && \text{(by IH)} \end{aligned}$$

Because $k > 3$, we know that $k+1 > 2$.
($k+1 > 4$ is also true, but we this is all we need.)

$$\begin{aligned} k! \cdot 2 &< k! \cdot (k+1) && \text{(because } k+1 > 2\text{)} \\ &= (k+1)! \end{aligned}$$

Therefore $2^{k+1} < (k+1)!$. □

You can also use this trick to make some computations that are way too nasty to do the long way (you can even do computations by hand that your calculator would choke on). And this isn't just when your input values are large. When you increase the number of numbers that you're multiplying together, the

[57] Eventually we will prove that the formula is true for the next value, $k+1$, but for now, we'll just look at the numbers involved in that formula by themselves, without assuming anything about how they're related to each other.

outcomes get big very quickly. By the time you get to double digit inputs, the factorials already have 6 digits: $10! = 362880$. $15! = 1307674368000$ is more than a trillion and too big for my cell phone's calculator program to handle. $20! = 2432902008176640000$ has 19 digits (over 2 quintillion already). 30! sends the Windows calculator program into scientific notation (meaning you're only getting an approximation[58]. My pocket calculator (which is powerful enough that I've nicknamed it Captain Overkill) is smart enough not to even try to compute 500!. It leaves it in symbolic form, which isn't very useful if you wanted to know what number that was (it's far bigger than you can imagine, and that's no exaggeration), but is useful if you wanted to figure out a value like $\frac{500!}{499!}$, which is surprisingly simple.

Example 5.32. Compute $\frac{500!}{499!}$.
$\frac{500!}{499!} = \frac{1\cdot 2\cdot\dots\cdot 499\cdot 500}{499!} = \frac{499!\cdot 500}{499!} = 500$

5.7 Summation Notation

Summation notation is even more versatile than factorials. It gives you a way to describe just about any sequence of repeated addition that you can dream up. All you need is a formula for how to figure out what you're going to add in the ith step. If you have such a formula, then you put a large capital sigma Σ (it's a Greek "S", for "sum") in front of the formula, put the starting point (in a form like $i = 0$ or $i = 4$) under the Σ and put the last value on top of the Σ (you can leave out the $i =$ here).

So if you wanted to represent $1 + 3 + 5 + 7 + 9 + 11$ in summation notation, you would come up with a formula to represent each term. In this case, the ith term can be written $2i + 1$ (provided you start with $i = 0$). To see this, check: $2 \cdot 0 + 1 = 1$, $2 \cdot 1 + 1 = 3$, $2 \cdot 2 + 1 = 5$, etc. We're starting at $i = 0$ and there are 6 terms in the sum, so we would write this sum as $\sum\limits_{i=1}^{6}(2i + 1)$. Why is this better than writing $1+3+5+7+9+11$? It's not, really. But often we'll want to do something like this without knowing exactly how many terms there will be. For example, yesterday, we wrote down (and proved correct!) a formula for the sum of the first n positive integers which worked no matter what n we picked. Instead of writing $1 + 2 + 3 + \cdots + n = \frac{n(n+1)}{2}$, which depends on people being able to guess what the pattern in the $\cdots$ is (this time it's easy) and is confusing when $n \leq 3$, we can write $\sum\limits_{i=1}^{n} i = \frac{n(n+1)}{2}$.

I'm sure this looks much worse to you right now, but this kind of representation appears everywhere in mathematics. It lets you reason about repeatedly applying any operation, even when you don't know how many times it needs to be done.[59]

[58] In most of the places where factorials are used, intermediate approximations like this are *very* bad.

[59] For example, if you wanted to do repeated multiplication instead of addition, you write things in exactly the same way, only you use a capital pi Π (the Greek "P" for "product"). It's also done for repeatedly unioning or intersecting sets, with a big $\cup$ or $\cap$.

Example 5.33. Compute $\sum_{i=0}^{4}(2i+1)$.

$\sum_{i=0}^{4}(2i+1) = (2\cdot 0+1)+(2\cdot 1+1)+\cdots+(2\cdot 4+1) = 1+3+5+7+9 = 25$

Example 5.34. Compute $\sum_{i=0}^{5}(2i+1)$.

$\sum_{i=0}^{5}(2i+1) = (2\cdot 0+1)+(2\cdot 1+1)+\cdots+(2\cdot 4+1)+(2\cdot 5+1)$. Notice that this is the same as the previous example (adding up term 0 through term 4) and then adding term 5, which is 11. In other words, $\sum_{i=0}^{5}(2i+1) = \sum_{i=0}^{4}(2i+1)+(2\cdot 5+1) = 25+11 = 36$.

This connection between a sum of n terms and a sum of $n-1$ terms is exactly what makes induction usable for sums. Usually, you'll have made some assumption (the induction hypothesis) about the sum of the first k terms (something like $\sum_{i=1}^{k} i = \frac{k(k+1)}{2}$), and you'll want to prove a similar fact about the sum of the first $k+1$ terms (like $\sum_{i=1}^{k+1} i = \frac{(k+1)(k+1+1)}{2}$). So you write the sum of the first $k+1$ terms as the sum of the first k terms plus the $(k+1)$th term (the last one), and then you can use what you know about the sum of the first k terms (in this case, $\sum_{i=1}^{k+1} i = \left(\sum_{i=1}^{k} i\right) + (k+1)$).

Example 5.35. Prove that for any $n > 0$, $\sum_{i=1}^{n}(2i-1) = n^2$

Proof. (Induct on n)
(Base step $n = 1$):

$\sum_{i=1}^{1}(2i-1) = 2\cdot 1 - 1 = 1 = 1^2$

(Induction step):

Assume $\sum_{i=1}^{k}(2i-1) = k^2$ for some $k > 0$.

$\sum_{i=1}^{k+1}(2i-1) = \left(\sum_{i=1}^{k}(2i-1)\right) + 2(k+1) - 1 = k^2 + 2(k+1) - 1$ (by the induction hypothesis)

$k^2 + 2(k+1) - 1 = k^2 + 2k + 2 - 1 = k^2 + 2k + 1 = (k+1)^2$ □

Example 5.36. Prove that for any $n \geq 0$, $\sum_{i=0}^{n} \frac{1}{2^i} = 2 - \frac{1}{2^n}$

Proof. (Induct on n)
(Base step $n = 0$):

$\sum\limits_{i=0}^{0} \frac{1}{2^i} = 2^0 = 1 = 2 - \frac{1}{1} = 2 - \frac{1}{2^0}$

(Induction step):

Assume $\sum\limits_{i=0}^{k} \frac{1}{2^i} = 2 - \frac{1}{2^k}$ for some $k \geq 0$.

(Goal: $\sum\limits_{i=0}^{k+1} \frac{1}{2^i} = 2 - \frac{1}{2^{k+1}}$)

$$\begin{aligned} \sum_{i=0}^{k+1} \frac{1}{2^i} &= \left(\sum_{i=0}^{k} + \frac{1}{2^i} \right) + \frac{1}{2^{k+1}} \\ &= 2 - \frac{1}{2^k} + \frac{1}{2^{k+1}} && \text{(by IH)} \end{aligned}$$

(We want to show that this $(2 - \frac{1}{2^k} + \frac{1}{2^{k+1}})$ is the same as $2 - \frac{1}{2^{k+1}}$. The 2's match up already, so let's take the rest $(-\frac{1}{2^k} + \frac{1}{2^{k+1}})$ and write it as one fraction, hoping that it will match $-\frac{1}{2^{k+1}}$. I'm writing more steps than are needed here, just to be extra clear.)

$$\begin{aligned} 2 - \frac{1}{2^k} + \frac{1}{2^{k+1}} &= 2 - \frac{2}{2^k \cdot 2} + \frac{1}{2^{k+1}} \\ &= 2 - \frac{2}{2^{k+1}} + \frac{1}{2^{k+1}} \\ &= 2 + \frac{-2+1}{2^{k+1}} \\ &= 2 + \frac{-1}{2^{k+1}} \\ &= 2 - \frac{1}{2^{k+1}} \end{aligned}$$

Therefore $\sum\limits_{i=0}^{k+1} \frac{1}{2^i} = 2 - \frac{1}{2^{k+1}}$. □

5.8 More Examples

Example 5.37. Prove that for any $n > 0$, $n^3 - n$ is divisible by 3.

General strategy: To show some number (say 12) is divisible by another number (say 3), rewrite the first number as the second number times some other integer $(12 = 3 \cdot 4)$. For example, to show that $5m^2 + 10$ is divisible by 5 (provided that m is an integer), I would write $5m^2 + 10 = 5 \cdot (m^2 + 2)$, and that would be enough because $m^2 + 2$ is an integer. If you already know (because it was an assumption, maybe) that some number (say $m + 1$) is divisible by another number (say 4), then you know you can write the first number as some integer times the second number, but you might not know what that integer is. In this case, you can give this mysterious integer a name $(m + 1 = j \cdot 4$ for some integer $j)$, and then work with that.

Proof. (induction on n)
(Base Step, $n = 1$):
$1^3 - 1 = 0$, and 0 is divisible by 3 because $0 = 3 \cdot 0$.

(Induction Step):
Assume that $k^3 - k$ is divisible by 3 for some $k > 0$. This means that $k^3 - k = 3j$ for some integer j.
(Will show: $(k+1)^3 - (k+1)$ is divisible by 3.

$$\begin{aligned}
(k+1)^3 - (k+1) &= (k^2 + 2k + 1)(k+1) - k - 1 \\
&= (k^3 + 3k^2 + 3k + 1) - k - 1 \\
&= k^3 + 3k^2 + 3k - k \\
&= k^3 - k + 3k^2 + 3k \\
&= 3j + 3k^2 + 3k && \text{(by IH)} \\
&= 3(j + k^2 + k)
\end{aligned}$$

Since $j + k^2 + k$ is an integer, this means that $(k+1)^3 - (k+1)$ is divisible by 3. □

Example 5.38. Any set with n elements has has 2^n subsets.

This is a tricky proof, but not because of the induction. I won't expect you to generate a proof like this on an assignment, but you should be able to understand most of this, and I think it's important that you see at least one example that's not just about arithmetic. If you have trouble understanding the reasoning in the middle of the third paragraph of the induction step, that's okay. I hope you'll be able to follow the rest of it. If nothing else, you should be able to understand the overall structure: The base step is a simple computation about the empty set (the only set with k elements). The induction step begins with assuming that we've already proven the formula for any set with k elements. We look at a set with $k+1$ elements and relate the number of its subsets to the number of subsets of a smaller set (one with k elements). This smaller set has k elements, so we can use the induction hypothesis to get a formula for the number of its subsets. Then finally, we use this computation to prove that the formula holds for the bigger set with $k+1$ elements.

Proof. (Induct on n, the number of elements in a set.)
(Base step $n = 0$, i.e., the set has no elements):
Any set with no elements must be the empty set. The only subset of the empty set is the empty set itself. So if a set has 0 elements, it only has 1 subset (which agrees with our formula 2^0).

(Induction step):
Assume that *every* set with k elements has 2^k subsets for a particular $k \geq 0$.

Suppose we have a set with $k+1$ elements, and call it A. (I'm going to show that A has 2^{k+1} subsets.) I'll give the names $x_1, x_2, \ldots, x_{k+1}$ to the elements of A so that we can talk about it. In other words $A = \{x_1, x_2, \ldots, x_{k+1}\}$.

Now some subsets of A have the element x_{k+1} in them, and some don't. We can count how many subsets of each type there are and then add them up to get how many subsets A there are. Actually, it's even easier than that, because you can match up the subsets that don't have x_{k+1} with those that do. For example, the subset $\{x_2, x_5, x_{k+1}\}$ matches up with the subset $\{x_2, x_5\}$. All the subsets can be paired up like this, so the two counts (subsets with x_{k+1} and subsets without x_{k+1}) must be the same. So instead of counting both types of sets, we'll just count one type and double that count to get the total number of subsets in A.

Let's look at the subsets that *don't* have x_{k+1} in them. The subsets of $\{x_1, x_2, \ldots, x_{k+1}\}$ that don't have x_{k+1} in them are really just the subsets of $\{x_1, x_2, \ldots, x_k\}$. But $\{x_1, x_2, \ldots, x_k\}$ has k elements in it, so it's a special case of the induction hypothesis. By our induction hypothesis, $\{x_1, x_2, \ldots, x_k\}$ must have 2^k subsets. So there are 2^k subsets of A that don't have x_{k+1} in them.

So there are 2^k subsets that have x_{k+1}, and by our earlier argument, there must also be 2^k subsets that don't have x_{k+1}. Therefore, the total number of subsets in A is $2^k + 2^k = 2 \cdot 2^k = 2^{k+1}$. □