Philosophy 110: Introduction to Logic

Daniel Hadad

Last Updated 5/16/2021

1 Unit 1: Sentential Logic

1.1 What's the Point of Studying Logic?

People take logic classes for a variety of different reasons. Unfortunately, for many the reason is that they are required to take it in order to graduate. Often, students will take a logic course in order to avoid having to take a class in mathematics, a subject that they either hate, or find too difficult (maybe that's *why* they hate it).

I have some good news and some bad news. I'll start with the bad news: if you can't stand math, you probably won't enjoy formal logic too much either.

The good news is that there are many reasons why studying logic can be exciting and enjoyable, if you come into it with an open mind. In fact, it gets even better: doing well in this course will make math seem a lot *less* daunting in the long run.

I won't beat you over the head with my admiration for formal logic, but I will mention a few aspects of it that I think are really amazing.

- Formal logic makes your computer work. No, really. Your computer isn't anything more than a big box that executes millions of logical operations one after another. I like to point this out to skeptical students, the way that some math students ask "What is calculus even good for??". The truth is that if the logicians working at the turn of the 20th century (people like Alan Turing, for instance) weren't so interested in formal logic, you wouldn't have a laptop or a smartphone today.

- Learning logic can greatly improve the speed and clarity of your thoughts. Even if you're not interested in building computers, or getting better at math, or whatever else, learning logic can improve *how* you think by making you a better abstract thinker. If you find yourself struggling to hold information in your head, or to consider hypothetical situations and possibilities, then formal logic (and philosophy more generally) can really change your life.

Well, that's my plug for learning logic. If it hasn't done anything to sway your opinion towards being excited to take this course, I'll leave you with one final piece of advice:

Treat everything in this class like you're playing a game. I'll be giving you a number of rules for how to play, and then various problems that require you to apply those rules. So if you ever lose sight of *why* you're doing something, don't get discouraged. Just think of it all like a big game, where you win points by answering the questions correctly, and your hopeful reward is an 'A'.

1.2 The Features of an Argument

Logic is basically the formal study of **arguments**. Now, when logicians say argument they don't mean a disagreement or quarrel between people. Instead, they have something more technical in mind.

Definition: An **argument** is a set of statements where some of the statements (premises) are put forth as *supporting* (or justifying, or providing a reason/reasons for) another statement (conclusion).

Unfortunately, that definition doesn't do us much good right now. We can greatly improve it by providing some other definitions.

Definition: A **statement** is a sentence, or part of a sentence, that is either *true* or *false*, i.e. that has a *truth value.*

For example, the sentence "The Toronto raptors won the 2019 NBA championship." is a statement, because it's a sentence that's either true or false (in this case, it's true).

Similarly, the sentence "Today is the 12th of June, 1923" is also a statement for the same reason (but this time, it's false).

On the other hand, sentences that are either *questions* or *commands* are not statements, since they're neither true nor false.

E.g. "Go close the door!" ; "Have you visited your grandmother recently?"

Lastly, note that a sentence can contain multiple statements at the same time.

E.g. "John's favorite color is blue, but he doesn't fancy the blue sweater that Jane got him because he thinks it's itchy." contains the following statements:

- "John's favorite color is blue."
- "John doesn't fancy the blue sweater that Jane got him."
- "John thinks the blue sweater that Jane got him is itchy."
- And finally, the whole sentence itself is a statement.

In philosophy, it is common to make a distinction between a *statement* and a *proposition*, which is the **meaning** of the statement. For instance, we can use this distinction to easily show that "Snow is white" and "Schnee ist weiß" express the same proposition using different statements (because one is a sentence in English and the other is in German). However, for the purposes of our class in logic, we will sometimes use these words interchangeably. If the distinction is ever relevant to the discussion, I'll make a note of it.

Now that we have a clearer definition of a statement, we can define the two kinds of statements that make up an argument in our more technical sense of the word.

Definition: A **premise** is a statement put forward in an argument that *provides support or justification* for another statement.

Definition: A **conclusion** is a statement in an argument that *is supported or justified* by the premises of the argument. Commonly, 'the' conclusion of an argument is the proposition that the argument aims to prove or convince the listener to believe.

Premises and **conclusions** are the building blocks of philosophical arguments. In the coming sections, we will see some basic examples in English of how arguments are laid out, and describe a crucial notion: *deductive validity.*

1.3 Deductive Reasoning, Validity, and Soundness

1.3.1 Deductive and Inductive Reasoning

In our everyday lives, we are constantly engaged in reasoning about the world around us. Any time you make a conscious decision, weigh the pros and cons of doing something, make an educated guess on a multiple choice exam, and so on, you're reasoning. This reasoning can take different forms. The study of logic is primarily concerned with a kind of reasoning called **deductive reasoning**.

Definition: A **deductive argument** is one in which the conclusion is put forward as following *necessarily* from the premises, i.e. if the premises are true then the conclusion *must* be true.

Don't worry too much about this definition at the moment. We'll work through some examples in just a bit to better demonstrate what this definition means. But before we do that, let's talk a bit about a common kind of reasoning that we're *not* concerned with in this course.

Definition: An **inductive argument** is one in which the conclusion is put forward as *probably* following from the premises, i.e. if the premises are true, then the conclusion is *probably* or *most likely* true.

We use inductive reasoning all the time. It's the sort of reasoning that comes into play when you notice **trends** or **patterns**. Think about ordering your usual meal at your favourite restaurant. Before you place the order, you believe that you will enjoy the meal. However, there's no guarantee that this is true. The chef could be having an off night and make a mistake with your order, for instance. But based on your past experiences, you think it's probably going to be good.

The same kind of reasoning applies to things like weather forecasts. If the weather app on your phone says there's an 80% chance of rain, then you believe that it will most likely rain that day, even though 20% of the time the day will be rain-free. Even with good inductive reasoning, it's possible for the all premises to be **true**, but for the conclusion to be **false**. You can have justified beliefs about how things will turn out, even if chance proves you wrong.

With **deductive reasoning**, on the other hand, this is not possible. Let's take a look at a very basic kind of deductive argument, called a **syllogism**, to better see this distinction. Take a note of how this argument is written (commonly referred to as 'standard form'), since we will be using this format throughout the entirety of the course.

1. All humans are mortal.
2. Socrates is a human.

$\therefore$ 3. Socrates is mortal.[1]

[1] The symbol '$\therefore$' means 'therefore'.

The way that we read an argument like this is that statements 1. and 2. are the *premises* and 3. is the *conclusion*. At this stage, you might not as yet see the way in which this argument is special or interesting, but it has an important feature that we will be focusing on in this course: it is **deductively valid**.

> *Definition:* An argument is **deductively valid** if the truth of the premises entails the truth of the conclusion. A valid argument *cannot* have all true premises and a false conclusion. Whenever the premises of a valid argument are all true, the conclusion *must* be true. If it is possible for an argument to have all true premises and a false conclusion, then the argument is **invalid**.

It is easy enough to show that our basic argument is valid. If it's true that all men have the property of being mortal, and it's also true that Socrates has the property of being a man, then Socrates has got to be a mortal! There's no way for the premises to be true without the conclusion also being true, and that's just what it means for an argument to be valid.

1.3.2 Truth and Validity

Deductive validity is somewhat of the centerpiece of an introductory course in logic. One way to summarize the goal of a course like this is to understand the formal system of rules that governs deductively valid arguments. In order to do that, we need to have a very firm understanding of validity.

The first thing that needs to be stressed is that validity is a formal or technical feature of an argument's *structure*. An argument is valid in virtue of its *form*, not its particular *content*. This means that arguments can be valid even if their premises are false. Arguments can also be invalid even though the premises *and* and the conclusion are all true. The only thing that isn't possible is a valid argument with *all true premises* and a *false* conclusion. Without a few examples, this might be a bit difficult to understand, so let's go through the possibilities.

E.g. A valid argument with a false premise and a false conclusion.

1. All birds can fly.
2. Penguins are birds.

$\therefore$ 3. Penguins can fly.

This argument is valid. In fact, you might have noticed that it has the exact same logical structure as the earlier argument regarding Socrates.[2] If it's true that all birds have the ability to fly, and that penguins are birds, then it *must be true* that penguins have the ability to fly. Of course, it isn't actually true

[2] If this isn't clicking as yet, don't worry. We will talk a lot more about logical form later on in this Unit.

that all birds can fly, but ***if it were true*** then the conclusion of the argument would be guaranteed to be true, and that's all an argument needs to be valid.

E.g.2 An invalid argument with true premises and a true conclusion.

1. Some dogs are friendly.
2. Some dogs are brown.

$\therefore$ 3. Some dogs are friendly and brown.

This is an example of an invalid argument. All of the statements that make up the argument are true, but I'm saying that it is a bad argument. Why? Well because it doesn't have the right structure. The reasons given (the premises) don't guarantee that the conclusion is true. The conclusion *happens to be* true, but not *because* the premises are true. If you're not already convinced, let's look at one more example.

E.g.3 A counterexample showing the invalidity of the argument in E.g. 2

1. Some dogs weigh more than 60 pounds.
2. Some dogs weigh less than 60 pounds.

$\therefore$ 3. Some dogs weigh more than 60 pounds and less than 60 pounds.

This argument has the exact same structure as the previous one. The first premise says that some dogs have a particular feature. The second premise says that they have another, different feature. The conclusion says that they have both of the features mentioned at the same time. The difference is that the features in E.g. 2 are both possible for the same dog to have, but in E.g. 3 this is not the case – no dog can weigh both more and less than 60 pounds at the same time! And that shows the problem with the original argument in E.g. 2. If *that* argument were a good one just in virtue of its structure, then the argument in E.g. 3 would also be a good argument. The argument in E.g. 3 is obviously not a good argument, though, because its conclusion is impossible.

One way to think about it is like this: even though it's true that some dogs are both friendly and brown, we can't use the reasons that some dogs are friendly and some dogs are brown to prove that, because it's possible (as E.g. 3 demonstrates) for those two premises to be true while the conclusion is still false. The counterexample is a way to see that in effect.

An argument cannot be logically convincing if it is invalid. Validity – like the carefully laid foundation of a home – is necessary for an argument to be logically convincing. It's no surprise, then, that the majority of this course is concerned with understanding methods for determining valid and invalid forms of reasoning.

When evaluating an argument, we can only turn to whether the premises are true after we've determined that the argument is valid. At this point, you might imagine that the 'gold standard' for giving an argument is one that is both valid and has all true premises. This is what we should strive for when constructing our arguments in our day-to-day reasoning. In philosophy, we call these arguments 'sound'.

Definition: An argument is **sound** if and only if it is both deductively valid and has all true premises.

Our course is a course on logic, so we are not as concerned with soundness as we are with deductive validity. The goal of developing a system of logic is to determine what constitutes valid reason – logicians leave investigation of what the world is like to others. But despite its abstract nature, logic is extremely powerful as a tool of thinking and reasoning. In the rest of the course, we will develop a great understanding of the formal system of classical logic.

1.4 Some Axioms of Classical Logic

Armed with a good understanding of deductive validity, we are now in a position to begin laying out the formal aspects of the system of classical logic. These formal concepts will allow us to start modeling our natural language of English in a way that is more rigorous. This rigor affords us the opportunity to see the logical structure of the language much more clearly. Here are three important *axioms*, or *fundamental rules* of our system of logic:

Axiom 1: Classical logic is **two-valued**, which means that there are only two **truth values**. These are 'true' and 'false'.

Axiom 2: Classical logic obeys the **Law of the Excluded Middle**: for any proposition in the language, that proposition is either true or false. In other words, **every proposition must have a truth value**.

Axiom 3: Classical logic obeys the **Law of Non-contradiction**: No proposition in the language can be both true and false. In other words, **every proposition has *exactly* one truth value**.

These three axioms give us a good basis for our logical system, but they cannot do all the work for us. For instance, none of these axioms tell us anything about when we can make valid inferences from one set of propositions to another. Think back to our earlier examples of valid and invalid arguments. Each of the premises in those examples satisfied the axioms, but that alone could not tell us which arguments were valid. In order to do that, we needed to investigate the *structure* of the arguments. The rest of Unit 1 explains some of the tools that logicians use to evaluate these logical structures.

1.5 Atomic and Compound Sentences

Our first foray into formal logic involves what's called *sentential logic*. What this means is that individual statements will be the 'smallest' unit of language that we will consider in this Unit. In order to do this, we need to make a distinction between an *atomic* sentence and a *compound* sentence.
In English, it is common to conjoin sentences to one another using words like 'or' or 'and'. Using these words, we can 'build up' more complicated statements from simpler ones. Recall an example from **Section 1.2**:

E.g. "John's favorite color is blue, but he doesn't fancy the blue sweater that Jane got him because he thinks it's itchy."

This sentence is a combination of several smaller sentences, with the words 'but' and 'because' joining the individual sentences together. On the other hand, the sentence "John's favourite color is blue" lacks any such connecting words. Thus, we call the longer, more complex sentence a **compound statement**, while the unconnected one is an **atomic statement**.

Definition: An **atomic statement** in one that contains no connecting words or phrases. It is the smallest unit of sentential logic.

Definition: A compound statement is one that contains at least one atomic statement with at least one connecting word or phrase.

In the remainder of the unit, we are going to develop a basic technical foundation for understanding how we can ascertain the truth values of compound statements from the truth values of atomic ones. In order to do this, we need to understand the concept of a **truth-functional connective**.

1.6 Truth-Functional Connectives

Suppose you're on a crowded bus, and you overhear one passenger say to another that "Agnes likes rock climbing". The other passenger then replies "She likes baking as well". As we have just discussed, these are two **atomic statements**. Now consider the compound statement "Agnes likes rock climbing *and* she likes baking". Is this compound statement **true** or **false**? I think we would all agree that it is true. This suggests that two true statements joined together by the word ‘and’ form one compound statement that is also true.

Now suppose instead that you overhear a conversation where one person says “The Magna Carter was signed in 1215”, and another person says “The U.S. Declaration of Independence was signed in 1776”. Those are both true. But what about the sentence “The U.S. Declaration of Independence was signed in 1776, *before* the Magna Carta was signed in 1215”? Uh oh! This compound statement has the order of events wrong, and so it ends up **false**, even though its atomic components are both true. If we swapped the order to go the right way round, however, then the compound statement would be true, using *exactly* the same atomic components. In other words, whether a complex sentence that uses the word ‘before’ is true or false depends on more than just the truth values of its atomic parts.

What we are seeing in action here are one example each of a **truth-functional connective** (in the case of ‘and’), and a **non-truth-functional connective** (in the case of ‘before’).

Definition: A **truth-functional connective** takes the truth-values of atomic statements as *inputs* and gives the truth value of complex statements as an output. The truth value of a complex statement that contains a truth-functional connective is *solely determined* by the truth values of the atomic statements that make it up.

Truth-functional connectives are of paramount importance to our study of logic, because they allow us to see how the truth-values of complex statements get built up by the truth values of their atomic parts. Thus, we can use truth-functional connectives to translate sentences in English into a clearer logical notation that reveals the logical form of these statements. This makes it much simpler for us to determine what kinds of inferences are deductively valid – which is, as you might recall, our main aim.

There are a few ways to describe the total set of truth-functional connectives in English. At the end of the unit, we’ll talk a little about why logicians favor one set over another. For now, though, you should take my word for it that the five truth-functional connectives best suited to modeling the English language in logic are:

- ‘not’ (‘$\neg$’)
- ‘and’ (‘$\wedge$’)
- ‘or’ (‘$\vee$’)
- ‘if...then’ (‘$\implies$’)

- 'if and only if' (' $\iff$ ')

1.7 Constants and Variables

Before we start looking at any of the connectives in particular, we need to do some housekeeping about the terminology that will come up in the remainder of the unit. Mainly, we're going to be doing two sorts of things: (1) translating English sentences into logical notation; and (2) using **truth tables** [3] to visualize the logical meaning of the connectives, and analyze more complex statements for validity.

In order to make this process easy and concise, logicians rely on abbreviations and technical notation. For our purposes, we will focus on two sorts: constants and variables.

Definition: A **constant** is a symbol used to refer to a *particular* statement in English.

E.g. We might use the letter 'A' to refer to the statement 'Ann is eighteen' in a particular situation or context. We would write 'A = 'Ann is eighteen'', so that throughout that particular discussion we can just write the letter 'A' instead of the full sentence over and over.

On the other hand, we sometimes rely on the use of **variables**.

Definition: A **variable** is a symbol used to represent statements *in general* without referring to a specific statement.

We will see many examples of variables in the coming sections. The important distinction to keep in mind is that variables do not have a truth value, since a variable is just a symbolic stand-in for statements in general. However, constant symbols have specific meanings, since we use those to replace actual English sentences, and so every constant will have a defined truth value.

We will use the following convention throughout the course:

- For **constants**, use uppercase letters (usually ones that correspond with the proper nouns used in the sentences, though it does not matter much).
- For **variables**, we always use lowercase letters starting with 'p' for the first variable, 'q' for the second, and so on.

[3] Explanation just around the corner!

1.8 Logical Negation: The Truth-functional Connective 'Not'

The first connective that we will be looking at is 'not'. The logical operator is called 'negation', and it is symbolized by either '$\neg$' or ' '. I'll be using the former.

Negation changes the truth value of a statement from true to false, and vice versa. If you negate a true statement, you get a false one. If you negate a false one, you get a true one. Let's do a couple of examples, and meanwhile let's practice using the symbols mentioned in **Section 1.7**.

E.g. We consider the statement "James is an excellent swimmer", which we will symbolize as J. Now suppose we want to write the statement "James is *not* an excellent swimmer" (or, as I'll explain in just a bit "It is not the case that James is an excellent swimmer"). In this case, we can simply write '$\neg J$' [4], successfully translating the negation of our original statement into logical notation.

1.8.1 A Note on Translating Negations

As you can see, translating the negations of statements into logical notation isn't that hard. However, there are some difficulties that come up when we realize that English is a messier language than logicians would like it to be.

For instance, let's look at the statement "Deidre is good at maths." What would the negation of this statement look like in English? You might think that these would all do the trick:

- "Diedre is *not* good at maths."
- "Diedre is *bad* at maths."
- "It is not the case that Diedre is good at maths."

If you use the constant 'D' = "Deidre is good at maths.", then each of those three options will get translated as '$\neg D$'. In this case, those three statements all share the same meaning.

But now let's think about the sentence "Everyone in the class is happy" = 'E'. How would we go about writing '$\neg E$' in English? Here are some options:

- "Everyone in the class is *sad*."
- "Everyone in the class is *not* happy."
- "*Not* everyone in the class is happy."

[4] Pronounced 'Not J'

- "It is not the case that everyone in the class is happy."

Do you see the problem? The first two are definitely no good! If it's false that everyone in the class is happy, that doesn't mean that *everyone is sad*. Similarly, it doesn't necessarily mean that *everyone* is not happy. We will have to throw those options out.

The last two do a much better job of translating what we intuitively mean when we say that E is false, or that $\neg E$ is true.

Logicians call the kind of mistake that occurred in the first two translations a 'scope' issue. In sentential logic, we always want negations to have what's called a 'wide' scope, which means that the entire sentence gets negated. In the case of D this did not cause us much difficulty. But with E we ran the risk of saying something much different than what we intended. This potential for confusion has caused logicians to adopt the phrase 'it is not the case that' to indicate negation. This helps us remember to always give the negation wide scope in our translations, thus making it trivial to translate negations from logical notation back into English. Remember to be careful about scope when translating sentences from English into logical notation!

1.8.2 The Truth Table for Negation

We have already captured the meaning of logical negation when we said "If a statement is true, its negation is false" and "If a statement is false, its negation is true". Logicians have developed a visual method for encoding the 'meaning' of a connective or set of connectives, called a 'truth table'. Truth tables are a wonderful tool for systematically evaluating the truth values of compound statements based on their atomic parts and the logical connectives that are present, so we will spend a lot of time familiarizing ourselves with how they work. Here is the truth table for negation [5]:

p	$\neg p$
T	F
F	T

Table 1: Truth Table for 'NOT'

In order to effectively use truth tables, we need to learn to read them. The first row of the truth table lists all of the statements that we are analyzing. Since this truth table is meant to encode the meaning of 'not' in logical notation, the left-hand column is a variable for any atomic statement, and the right column is the compound statement that you get from negating the atomic statement.

[5] Take notice of the fact that we're using variables here instead of constants, since we are trying to visualize the negation operator for statements in general.

N.B The top row of a truth table always contains the statements that are being discussed, starting from atomic statements on the left and moving to compound statements on the right.

The subsequent rows are reserved for the potential truth values of each of the statements. For example, the second row says that when p has the truth value 'true', $\neg p$ has the truth value 'false', which is just another way of saying that when a statement is true, its negation is false.

Since there is only one atomic statement present, this truth table only requires two rows of truth values to cover all of the possibilities. As a result, the third row completes the truth table.

Over the course of the term we will become very proficient at both reading and writing truth tables. However, the first step is to familiarize yourself with all of the truth tables for our five truth-functional connectives. **You should be able to draw these basic truth tables from memory alone**.

That covers our first truth-functional connective, as well as how to read and interpret truth tables. Before you move on to the next connective, stop and attempt the exercises relating to negation specifically.

1.9 Conjunction: The Truth-Functional Connective 'And'

The next truth-functional connective on our list is 'and'. It is very common in logic to refer to a compound statement as a **conjunction** when 'and' is the primary logical connective.

Definition: A **conjunction** is a compound statement that has 'and' as its main connective. The atomic statements that are conjoined by 'and' are called **conjuncts**.

E.g. The statement "Frances is a law student and she studied philosophy" is a **conjunction**, where "Frances is a law student" and "Frances studied philosophy" are its two **conjuncts**. [6]

1.9.1 The Truth Table for Conjunction

Hopefully you're already thinking about how conjunctions work as truth functions. Intuitively, if both conjuncts are true, then the whole conjunction should be true as well. But what if either or both of the conjuncts are false? Surely we

[6] Try to familiarize yourself with this terminology as the course goes on.

would not endorse the sentence "Earth is flat and Gary drives a Civic" even if we find out that Gary drives a Civic, since the first conjunct – "Earth is flat", is false! The same thing goes for when both conjuncts are false. Thus, 'and' has a fairly easy and natural logical interpretation: **conjunctions are true when both of their conjuncts are true, and false otherwise**.

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

Table 2: Truth Table for 'AND'

Did you guess that this is what the truth table would look like? Notice that a few things have changed compared to the truth table for negations. Because conjunctions necessarily involve at least two atomic statements, we need two different variables – 'p' and 'q' – to draw the truth table. This results in a truth table with four lines instead of two. Hopefully it is easy to see that these four lines cover all of the possibilities.

You might be curious about the symbol '$\wedge$' that is used to symbolize 'and'. This symbol is sometimes called a 'caret'. Some logic textbooks use the traditional '&' instead of the caret. Others use a dot called the 'dot operator'. It's mostly a matter of personal preference and consistent use. I like the caret the most, so that's what I'm using throughout. In your translations, truth tables, and proofs, you're free to use any of the common symbols, so long as you make a note of which one you're using and you always use the same one.

1.9.2 Translating Conjunctions in English

As we have seen, the conjunction has a very natural interpretation as a truth-functional connective – true when the conjuncts are both true, and false otherwise. However, English is full of words and phrases that have a similar function as 'and'. They are used in different conversational contexts, but from a *logical* perspective, they all translate the same way as 'and'.

E.g. "Hans is tough, but fair." $= H$

How would you translate H into logical notation? Let's think about the information that H is conveying to us. In the first place, it tells us that Hans is tough (Say $T =$ 'Hans is tough'). It also tells us that he is fair ($F =$ 'Hans is fair'). The word 'but' is used because these two aspects of Hans's personality contrast with one another – they're dissimilar, a fact that the speaker is drawing to the listener's attention.

What would make H true? Take a second to consider the possibilities. H is only going to be true when *both* T *and* F are true. If Hans is tough but unfair, then H will be false. Similarly, if Hans is not tough, but he is fair, then H will be false. This is even more obvious when both T and F are false.

What this means is that the conditions under which H is true are indistinguishable from the truth conditions for a conjunction! And that means that H *just is* a conjunction from a logical perspective. The word 'but' carries additional meaning, but we are only interested in the logical aspects of the sentence. So we can translate H into a conjunction that looks like '$T \wedge F$'.

N.B. The English word 'but' almost always functions in logic exactly like the word 'and'.

There are other words and phrases that work like 'and'. Consider the phrase 'even though', for instance, in the sentence 'Ingrid enjoyed the show, even though she dislikes going out at night'. This statement is a conjunction of two atomic statements, 'Ingrid enjoyed the show' and 'Ingrid dislikes going out at night'. Once more, the connecting phrase 'even though' is used to demonstrate the surprising nature of the two pieces of information together, but the compound statement is only true when *both* of the atomic statements are true. Despite how this statement might first appear, it's nothing more than a conjunction!

You will gain a lot of practice by doing translations, and see even more common words and phrases that are logically identical to 'and'.

1.9.3 An Unfortunate Use of 'And'

We just saw how other words and phrases in English can often play the same logical role as 'and'. Unfortunately, sometimes 'and' is used in English in a way that *isn't* truth-functional. It's worth flagging what these problem cases look like.

Consider the following statements:

- "Jane and Karen are intelligent."
- "Jane and Karen are friends."

Now let us think about translating these statements into logical notation. In the first case, the sentence is written in a shorthanded way. If we wrote it out fully, it would say "Jane is intelligent and Karen is intelligent." Thus, we could translate it as "$J \wedge K$".

The second statement is more of a problem, as you may have already guessed. You would rightly object if I told you that the intended meaning of that sentence is "Jane is a friend and Karen is a friend." The statement is clearly trying to express the fact that Jane and Karen are friends *with one another*. This means that it can't even be broken down into smaller parts!

Ahh! It's very frustrating to see English sentences like this just after learning about the conjunction as a logical operator. Luckily there is a quick rule of thumb that we can use to avoid making any errors. **When the word 'and' is used to describe *the relation* between two or more things, it is not being used as a truth-functional sentence connective.**

We will revisit troublesome sentences like this one in **Unit 3**, where we will learn how to provide a translation for it. But for now, keep that rule of thumb in mind, and don't translate 'and' as a conjunction when it's being used in this relational way.

1.10 Disjunction: The Truth-functional Connective 'Or'

At this point we have covered both negations ('not') and conjunctions ('and'). The next truth-functional connective that we'll be learning about is 'or'. Just like 'and' and the term 'conjunction', logicians have developed a bit of technical terminology for 'or' statements.

Definition: A **disjunction** is a compound statement that has 'or' as its main connective. The atomic statements conjoined by 'or' are called **disjuncts**.

So there you have it – 'conjunction/conjunct' for 'and', and 'disjunction/disjunct' for 'or'. It will take some time, but hopefully these terms will become second nature.

E.g. "Liam's birthday is in June or July" is a **disjunction** that is made up of the two **disjuncts** 'Liam's birthday is in June' and 'Liam's birthday is in July'.

1.10.1 The Truth Table for Disjunction

As is now the standard order, let's move on to the truth table that gives the truth conditions (and hence the logical meaning) of 'or'.

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

Table 3: Truth Table for 'OR'

As you can tell, 'or' is symbolized using '$\vee$', which is called a wedge or 'v'. It looks just like the caret that we use for conjunctions, but flipped upside down (now you can see why I like using the caret).

Does the truth table look as you expected? Many students are immediately drawn to the first row of the truth table, the one that says that for any statements p and q, '$p \vee q$' is true when both p is true and q is true.

You might say to yourself "Hey wait a minute! The example of a disjunction that you just gave us said "Liam's birthday is in June or July", but his birthday cannot be in *both* months at the same time! Is the truth table wrong?"

1.10.2 Inclusive and Exclusive Disjunctions

Well, not quite. It turns out that the two disjuncts 'Liam's birthday is in June' and 'Liam's birthday is in July' do not accurately capture the statements that make up "Liam's birthday is in June or July". We'll get to that in a minute. First, consider another natural-sounding 'or' statement.

E.g. "You need either $2.50 or a monthly pass in order to ride the bus."

Suppose you get to the bus and swipe your monthly pass, but then the driver asks you if you also have $2.50. You say that you do, and then the driver tells you that you're not allowed to ride the bus. That would be completely bizarre! Similarly, if someone is making you a cup of coffee and asks if you'd like "cream or sugar", they should be fine if you ask for both!

These examples demonstrate an important aspect of the word 'or' as it's used in English. In some cases, the 'or' is meant in an **inclusive** way – such a sentence is true if one or the other **or both** disjuncts are true, like our bus example.

On the other hand, sometimes 'or' statements are **exclusive** – one of the disjuncts has to be true for the whole disjunction to be true, but they cannot both be true at the same time, as in the birthday example. I stated earlier that this example was not adequately represented by the two disjuncts "Liam's birthday is in June" and "Liam's birthday is in July". Now that we understand the distinction between an inclusive and exclusive disjunction, we can see that another piece of information was missing. In order to fully capture the meaning of the original statement, we need to expand it to say explicitly what was implied, i.e. "Liam's birthday is in June or July, **but it is not in both June and July**".

Returning to the truth table, it should now be clear that it represents only the inclusive meaning. That's because logicians frequently make the stipulation that uses of 'or' are inclusive unless otherwise specified.[7] This isn't the last stipulation that we'll be seeing when it comes to the meanings of our main connectives, but it is one that is easily justified.

The idea here is that the the exclusive use of 'or' asserts something strictly stronger than the inclusive use. This is because the inclusive use means "One, or the other, **or both**", whereas the exclusive use means "One, or the other, **but not both**". The inclusive use is, well, more inclusive! As a result, it makes a lot of sense to treat the more inclusive use as the standard, and make it explicit when we're using the exclusive sense instead.

Fortunately for us, there is a very natural way to use both negation and conjunction to define the exclusive sense of disjunction. Take two statements, A and B. Let's try to translate the statement 'A or B, but not both' into our logical notation.

First, we need to say 'A or B', which is very easily done as '$A \vee B$'. Next, we need to come up with a way to express 'but not both'. Recall from the section on conjunctions that 'but' functions just like 'and'. 'Not' is clearly indicating a negation, and 'both' – referring to 'both A *and* B' – indicates another conjunction. So, putting that all together, we get:

'A or B, but not both' = $(A \vee B) \wedge \neg(A \wedge B)$

N.B. In general, for statements p and q, the exclusive sense of 'or' is given by $(p \vee q) \wedge \neg(p \wedge q)$.

The parentheses may be causing you some confusion at the moment, since we have not discussed how and when they should be used. For now, since you already know the translation of this particular statement, try to guess what rules are governing the placement of the parentheses. It's OK if you don't get

[7] As an interesting aside, logic is used often in making electrical circuits used for computing. In that setting, there is an explicitly defined exclusive 'or' in addition to the inclusive one.

it, because once we're done going through all of the basic connectives, we will talk at length about using parentheses to write more complex statements like this one. Once we have all the syntax under control, we will also use the truth table method to further justify our translation.

1.11 Conditionals: The Truth-functional Connective 'If...then'

Moving on to our next basic connective, we have the grammatical construction 'if...then'. Fair warning in advance, this is usually the connective that gives new students the most trouble. We will take our time in going through it. As with the earlier connectives, logicians use some special terminology to talk about these kinds of statements.[8]

> **Definition:** A **conditional** is an 'if...then' statement. The statement (or 'condition') that comes after the 'if' and before the 'then' is called the **antecedent** of the conditional. The statement that comes after the 'then' is called the **consequent**.

One easy way to remember these terms is that the antecedent comes before ('ante' means before) the consequent (which sounds like 'consequence', the result of the condition being met).

Conditionals show up all the time in our daily lives. The following are all common expressions:

- "If Mark does not study, then he won't pass the class."
- "If $2 + 5 = 7$ and $1 + 6 = 7$, then $2 + 5 = 1 + 6$."
- "If the dough had been left to rest, then it would have risen."

These statements all appear to have the same 'if...then' structure. However, the way that structure imparts meaning onto each of those statements is very different. The first statement expresses something that Marks needs to do to pass the class (study). The second describes a logical or mathematical relationship. Finally, the third says something different altogether, since its antecedent isn't something that actually happened, but instead something that *could have happened*. Only the first two can be evaluated in a truth-functional way. The final statement – and others like it – have proven extremely difficult for philosophers to analyze in a careful and rigorous way. Philosophers refer to conditionals like the first two as 'material' conditionals, while the last is called a 'subjunctive' conditional.

[8]Unfortunately there's nothing we can do to sidestep the terminology. As an instructor, I often wish that it were more feasible to leave these technical terms aside to avoid students having to commit so many new words to memory, but rest assured that your early work will be rewarded, as it will be much easier to talk about these kinds of statements in the future.

1.11.1 The Truth Table for the Material Conditional

Up to this point, the truth tables for the basic connectives have been uncontroversial. The meaning of logical negation follows from our axioms – if you negate something true, you get something false, and vice versa. Similarly, it was natural to think that a conjunction should only be true when its conjuncts are both true. When it came to disjunctions, there was a bit more work to do, since we needed to justify using the inclusive, rather than exclusive, 'or'. Nevertheless, that justification came easily and made a lot of sense.

Regrettably, things won't go so smoothly with the conditional. Let's start with as quintessential a conditional as we can get, "If Mark does not study, then he won't pass the class." We'll abbreviate the relevant statements as $\neg S$ = 'Mark does not study' and $\neg P$ = 'Mark won't pass the class'.[9] Therefore, our statement can be partially symbolized as 'If $\neg S$, then $\neg P$'. We use the symbol '$\implies$' to represent the conditional.[10] Therefore, 'If $\neg S$, then $\neg P$' is symbolized as '$\neg S \implies \neg P$'.

First we ask ourselves, "If it's true that Mark doesn't study, and it's true that he won't pass, would it be true to say that 'If he doesn't study, then he won't pass'?" The answer seems a resounding 'yes'! We wouldn't assert a conditional if we thought the antecedent were true but the consequent were false – we use conditionals to indicate that the antecedent being true entails that the consequent is true.

By that same reasoning, we would want to say that the second line of the truth table – where the antecedent is true but the consequent is false – indicates that the conditional is false. This can be justified by considering our example. If we find out that Mark did not study, but that he passed the class anyway, we would *not* assert that if he did not study, then he would not pass. This reasoning allows us to give the first half of the truth table as follows[11]:

[9] Can you see why I chose to use '$\neg S$' and '$\neg P$'? It's because each of the statements contain a 'not', and so they are more naturally translated as the *negations* of the statements 'Mark studies' and 'Mark will pass the class'.

[10] Some textbooks use a single-width arrow '$\rightarrow$'. Others use a symbol called a horseshoe, '$\supset$'.

[11] Keep in mind that this truth table uses constants instead of variables because we're talking about a particular set of statements!

$\neg S$	$\neg P$	$\neg S \implies \neg P$
T	T	T
T	F	F
F	T	
F	F	

Table 4: The First half of the Truth Table for the Material Conditional

So far, so good. Only two more lines to fill. Here's where things start to break down. In the last two lines of the truth table, the antecedent of the conditional is false. What are we supposed to say about an 'if..then' statement that starts off false?

This is where the disconnect between English and logic is seen most clearly. In English, we would never sincerely assert a conditional that we knew had a false antecedent. This is because speaking with one another in natural language is about communication, and there are rules for communicating that go beyond truth values. To use our example, it would seem strange for someone to remark "If Mark does not study, then he won't pass the class" if they already knew that Mark studies. But, despite that fact, maybe we want to say that the conditional is still...true?

Things get even stranger when you consider nonsensical conditionals, where antecedent and consequent don't seem connected in any way. Statements like "If Obama was president before Trump, then $2 + 2 = 4$". Both the antecedent and consequent there are true, but something about the conditional seems very wrong. Those two facts have nothing to do with each other! This is a big problem for us as well, because now even the first row of the truth table seems less justified.

We can't get away from these examples, however, because unlike the non-truth-functional use of 'and' that we previously encountered, these conditionals are using the same truth-functional connective as the one we are analyzing. Simply put, conditionals with false antecedents, and ones with nonsense implications are here to stay. And as logicians, we have to make a call about how to assign truth values to the conditional in all cases. This is what the final truth table looks like:

p	q	$p \implies q$
T	T	T
T	F	F
F	T	T
F	F	T

Table 5: Truth Table for the Material Conditional

Not what you expected? It wasn't for me, either. In fact, almost every new student of logic sits staring at this truth table scratching their heads. There is, however, a strong justification for this choice[12]. The issue is that an in-depth discussion of the truth table for the conditional requires quite a bit more technical machinery than we have been exposed to at the moment.

I can offer you this much. In this brief discussion, we have seen good reasons to worry about every single line of the truth table, *except* for the second one. If we can be sure of one thing, it is that a conditional with a true antecedent but a false consequent is **false**. Thus, logicians decided to make this feature the defining characteristic of the material conditional. **A material conditional is false if its antecedent is true and its consequent false, and true otherwise**.

You might find it comforting, at least, that this interpretation of the conditional doesn't have any real negative effects on our communication in English. After all, people don't frequently go around asserting conditionals with knowingly false antecedents, or suggesting nonsensical relationships between disconnected facts. Furthermore, the material conditional does capture the essential logical properties that we, as English speakers, expect of 'if...then' statements, as we will see in the next few sections.

1.11.2 Translating Conditionals in English

Conditionals in English arise in many different forms. In the following example, we'll use N = 'Nina is having a good time' and O = 'Nina is out with her friends'. Consider the following sentences:

(a) 'If Nina is out with her friends, then she's having a good time.'

(b) 'If Nina is out with her friends, she's having a good time.'

(c) 'Nina is having a good time if she is out with her friends.'

(d) 'Nina is out with her friends only if she is having a good time.'

[12] Though not an uncontroversial one! In fact, whole branches of logic have been developed just to try to understand the different kinds of conditionals that occur in English.

All of these sentences get translated the same way, as '$O \implies N$'. (a) has the exact form of the conditional. (b) loses the word 'then', but it should be clear why it's the same statement without it. (c) gets the order of the statements swapped, but notice that the antecedent is still the statement that comes directly after 'if'. This is crucial. Lastly, (d) uses the construction 'only if' instead of 'if...then'. Whenever you see this, remember that what comes directly after 'only if' is the consequent of the conditional.

Sometimes conditionals can appear even without an 'if...then' or 'only if' construction. For example,

(e) 'Whenever Nina is out with her friends, she's having a good time.'

(f) 'Any time Nina is out with her friends, she's having a good time.'

Both (e) and (f) could reasonably be interpreted as '$O \implies N$'. In **Unit 3** we will see that there is an even better way to translate sentences like these, but for now we will translate these statements as conditionals.

Now is good time to stop and try your hand at some translation problems focusing on conditionals. You might also want to read over this section, focusing on the final truth table and these recent examples of conditionals. Once you feel like you're in a good spot, we can move on to the last of the basic connectives, the biconditional.

1.12 Biconditionals: The Truth-functional Connective 'If and only If'

We're nearing the end of our stage setting. Just one more connective to go. The good news is that the biconditional is a lot more manageable than the material conditional[13].

1.12.1 The Truth Table for the Biconditional

The prefix 'bi' suggests that the biconditional is somehow **two** conditionals, and that's pretty much right. The idea behind the biconditional is that it describes an 'if...then' relationship that goes in both directions. One way to think of the biconditional is as a shorthand for expressing the idea that, for two statements p and q, '$p \implies q$' AND '$q \implies p$'. If we wanted to write purely in notation, we would say '$(p \implies q) \wedge (q \implies p)$'. The biconditional, then, is basically an equivalent way to express this relationship between two statements.

As for the meaning of the biconditional, recall that the defining feature of the conditional was that it was false when its antecdent was true and its consequent

[13]That's a bit of irony for sure!

false. The same feature applies to the biconditional, only now the conditional goes in both directions. This means that the biconditional is true when both of the atomic statements **share the same truth value**. In truth table form, it looks like:

p	q	$p \iff q$
T	T	T
T	F	F
F	T	F
F	F	T

Table 6: Truth Table for the Material Biconditional

1.12.2 Translating Biconditionals in English

Biconditionals are not very abundant in English. When they do appear, it is usually with the constructions 'if, and only if' (sometimes shortened to 'iff') or 'just in case'. Here are some examples (but even they don't sound very natural).

- 'Peter got that jacket at a good price if, and only if, he got a big discount'.
- 'It's safe to eat sodium chloride just in case it's safe to eat table salt' (because they're the same thing).

Needless to say, we are unlikely to run into the biconditional much when doing translations of English sentences. However, when we earnestly begin doing proofs we will see the biconditional again and again, because it provides the basis for the notion of **logical equivalence**. But I'm jumping the gun a bit.

We've reached the conclusion of our introduction to the truth-functional connectives. Before you move on, you should be able to draw the truth tables from memory, and you should be getting better at recognizing the typical English constructions that express them. In the next section, I will lay out the syntactic rules of our logical system. These rules will give us a systematic way to build up larger and more complex statements that combine several connectives at the same time. Once we have done that, we can start translating much more interesting sentences, as well as analyze complex statements using truth tables.

1.13 The Rules of Formation

Earlier, when discussing disjunctions, I argued that it better suited our purposes to define the inclusive disjunction as the basic connective, and build the definition of the exclusive disjunction up from the definitions of the inclusive disjunction, negation, and conjunction. At the end of that discussion, I gave the defintion of the exclusive disjunction as '$(p \vee q) \wedge \neg(p \wedge q)$', for any statements

p and q, which I told you was read out loud as "p or q, but not both p and q". I came up with this complex formula, knowing where to place the parentheses and so on, by following the **rules of formation for well-formed formulas**. We will use the following recursive definition.[14]

Definition: A **well-formed formula ('wff')** of sentential logic is any formula that consists of:

1. An atomic statement, given either by a capital letter for a known constant, or a common letter ('p', 'q', and so on) in the case of a variable;
2. If A is a wff, then $\neg A$ is a wff;
3. If A and B are wffs, then $(A \wedge B)$ is a wff;
4. If A and B are wffs, then $(A \vee B)$ is a wff;
5. If A and B are wffs, then $(A \implies B)$ is a wff;
6. If A and B are wffs, then $(A \iff B)$ is a wff;
7. Any construction that cannot be obtained just by repeated applications of rules 1-6 is **not** a wff.

What this definition tells us is that every well-formed sentence in the language is either an atomic statement, or a statement built up by connecting two (or one in the case of negation) well-formed statements using one of the basic logical connectives. And that's it.

Next, we need to understand the conventions concerning the use of parentheses, which are used to help keep complex sentences from being too difficult to parse. The rules for parentheses are only implicit in the definition of a wff, so it will benefit us to go through a few examples.

E.g. '$A \implies B \vee C$' is not a wff, because it's ambiguous between two distinct sentences, '$(A \implies B) \vee C$' and '$A \implies (B \vee C)$'. Without the addition of the parentheses, it is impossible to tell which sentence is meant. You will notice that in the rules of formation, all wffs joined by a logical connective (besides negation) have parentheses enclosing the entire sentence. This is our first rule of thumb.

N.B. When two wffs are connected by a logical connective, the result has to be enclosed in a set of parentheses.

The exception to this rule is when only one logical connective is present in the entire sentence. '$A \vee B$' is clear on its own, so we can omit the parentheses from the outside. In general, we always leave off the last set of parentheses enclosing

[14] Which means we build the definition up from repeated applications of rules, starting with an initial or base case.

the entire sentence. Therefore, we write '$(A \implies B) \vee C$' instead of writing '$((A \implies B) \vee C)$'. Next we have the rules for applying negations.

N.B. A negation always has the smallest scope, i.e. it always negates the smallest complete sentence to which it can be applied.

E.g. In the wff '$\neg A \wedge B$', the negation applies only to 'A'. This sentence reads 'Not A, and B'. If we wanted to write 'Not both A and B' in our logical notation, we would need to enclose the conjunction in parentheses as '$\neg(A \wedge B)$'. The parentheses tell us that the smallest complete sentence that the negation can be applied to is the whole conjunction.

We also don't have to enclose sentences with negations in parentheses to negate them further. We would write '$\neg\neg A$' instead of '$\neg(\neg A)$' if we wanted to write 'Not not-A'. In general, we only enclose a group of symbols in parentheses when they're joined by one of the other sentence connectives. When you're in doubt, look back at the rules of formation box above, and make sure your wff resembles one of the sentences given there.

If this all seems like a lot to keep track of, don't despair. There will be a lot of practice problems to help build up your skills, and when it comes to this sort of thing, practice makes perfect. Remember to think of it as a game with very clearly-defined rules. Don't be afraid to refer back to them whenever you're stuck.

1.13.1 Identifying the Main Connective

In this section, we will discuss the idea of the main sentence connective. This should help in nailing down the concepts required to translate complex sentences from English. At the end, I'll do a few working examples with explanations.

First off, go back to the definition of a wff for a second. Notice anything interesting? What I'm getting at here is the fact that every wff in our language is just an instance of a sentence that uses *exactly one* of our basic logical connectives. No matter what, every wff can be reduced to either a negation, conjunction, disjunction, conditional, or biconditional. This is another way of saying that every wff has a 'main' sentence connective.

Definition: A wff's **main connective** is the one that has the widest scope for that wff.

By the end of this section, we will have identified the main connectives for each of the following:

(i) $\neg((A \vee B) \wedge (\neg A \vee \neg B))$

(ii) $A \wedge (\neg B \implies C)$

(iii) $(A \wedge B) \implies (C \vee D)$

Before we go on, I encourage you to stop and look these over for a minute. Can you guess what the main connective is for each of these? Write your answers down before you continue reading the explanations.

(i) has a lot going on. However, the answer is actually very straightforward. Let's start reading left to right. The first connective that we encounter is the negation. The next thing to do is ask ourselves 'what's the scope of this negation'? Our rule of thumb for negations tells us that their scope is always the *smallest* complete sentence to which they apply. What is that in this case? Notice that the very next symbol after the negation is an open parenthesis, '('.

This tells us right away that the smallest complete sentence to which the negation applies is the one that's inside the completed parentheses. And where does that set of parentheses end? All the way at the end of the sentence!

Definition: A **set of parentheses** always consists of exactly one left-parenthesis, '(', and exactly one right-parenthesis, ')'. A set of parentheses is always determined by finding the shortest distance between a pair of left and right parentheses.

This definition is a bit difficult to grasp just by thinking about it, but you can always test things visually. Here is how I would do it for this example. When I see the left-parenthesis immediately after the negation, I move my eyes across the sentence looking for the closest right-parenthesis. But the very next symbol that I encounter is another left-parenthesis. This tells me (if the formula is written correctly) that the next right-parenthesis that I see finishes this pair first, because the 'distance' between them is shorter. Therefore, I know that the right-parenthesis after 'B' closes off '$(A \vee B)$'. The left-parenthesis just after the negation is still left open.

After repeating this process for the pair surrounding '$\neg A \vee \neg B$', we finally come to the right-parenthesis that closes off our initial pair. That's how we can tell that the negation covers the entire sentence. Going back to our definition of the main connective, we see that this negation actually has the widest scope for the sentence, so this whole statement is a negation. If we make the substitution $C = ((A \vee B) \wedge (\neg A \vee \neg B))$, this shows us that our sentence is $\neg C$.[15]

[15] Pay attention to this substitution. It tells us that sometimes individual letters can represent complex sentences by themselves! It's all a matter of keeping track of things for each

Let's apply this process to (ii) and (iii) as well. For (ii), we see 'A' followed by a conjunction. On the right side of the conjunction, we have a conditional expression enclosed in a set of parentheses. These parentheses tell us that the conditional inside of them only covers the statement letters within the parentheses. This means that the scope of the conditional only covers '$\neg B$' and 'C', whereas the conjunction covers the whole sentence. Therefore, the conjunction has a wider scope than the conditional, making it the main connective for this sentence.

In (iii), we have two sets of parentheses with a conditional in-between. As you may be able to guess by now, this would make this sentence a conditional, with '$(A \wedge B)$' as the antecedent and '$(C \vee D)$' as the consequent. The conditional is the main connective.

It will take some time to be able to automatically read complex formulas with the correct main connective every time. Let's take a quick detour back to some translations of English to get some practice in.

1.14 'Neither...nor'; 'Not both'; 'Unless'

Many sentences in English contain more than one logical operator. But up until now, we have stuck to thinking about one connective at a time. That is about to change. Now that we have a handle on the rules of formation, we can move on to learning how to translate some common English constructions that feature more than one connective.

1.14.1 'Neither...nor'

The first sentence construction we will look at is 'neither...nor'. For example, we can consider the sentence "The movie was neither compelling nor well-produced". In this case, C = 'The movie was compelling', and W = 'The movie was well-produced'. How do you think this translation should go?

The first step is to identify the connectives that are present. Both 'neither' and 'nor' signal negations. However, neither of these are the main sentence connective, because they each have a narrow scope, negating just the particular statements that immediately follow them. There is one more sentence connective present, however, since 'neither this nor that' means 'not this and not that', i.e. it is a conjunction of two negations. Thus, the main connective is a conjunction. This yields the translation '$\neg C \wedge \neg W$'.

---

and every problem. Writing things down helps a lot.

N.B. In general, a 'neither...nor' statement is translated as the conjunction of two negations.

1.14.2 'Not Both'

The construction 'not both' differs from 'neither...nor', but the difference can be hard to remember if you're not careful. It all depends on the scopes of the connectives in the sentence. As an example, we'll translate the sentence "It will not both rain and snow tomorrow." Let R = 'It will rain tomorrow', and S = 'It will snow tomorrow'.

In this example, notice the position of 'not'. It comes before the word 'both'. The word 'both' signals a conjunction. Therefore, in this example we have the scopes of the connectives inverted compared to our 'neither...nor' example. In this case, the negation covers the whole conjunction. Therefore, we would want to translate this sentence as '$\neg(R \wedge S)$'.

N.B. In general, a 'not both' statement is translated as the negation of a conjunction.

1.14.3 'Unless'

The final English construction that we'll talk about in this section is the word 'unless'. For this one we have the sentence "John will attend class tomorrow unless he is sick". Let J = 'John will attend class tomorrow' and S = 'John is sick'.

Let's think about what information the sentence is telling us. We learn that John will attend class except under the condition that he is sick. Thus, the sentence gives a conditional for when John will show up to class – any time he *isn't sick*. So we get the translation '$\neg S \implies J$' or, rephrased in English 'If John isn't sick, then he will attend class tomorrow'.

However, there is another way that we can go with this translation, because we can also consider the situation in which we arrive at class and notice that John isn't in attendance. Given that we know that he will attend unless he is sick, we are in a position to say that *if* he doesn't attend, *then* he must be sick. In other words, we can also translate this sentence using a different conditional, '$\neg J \implies S$'.

N.B. An 'unless' statement is translated as a conditional. The negation of the statement that comes directly after 'unless' is the antecedent[16], and the other statement given is the consequent. Alter-

[16] Pay attention to what the antecedent says, because it may already be negated. For

natively, you can swap the conditional around, this time negating the new antecedant and leaving the new consequent as originally written.

Interestingly, we have two different translations for the same sentence. However, if they are both accurate translations, then they must mean the exact same thing! And, in fact, they do. They are **contrapositives** of one another. In the next section, we will cover sentence forms in a more general way, and then move on to computing more complicated truth tables. At that time, we will use what we have learned to prove that these two conditionals are logically equivalent to one another!

1.15 Logical Form

Earlier in the unit, we learned about the distinction between constants and variables. To refresh your memory, a constant is an uppercase letter that we used to represent a particular statement, whereas a variable was a lowercase letter starting with 'p' that we used to represent a statement in general. This distinction is important – constants help us do logic more easily for particular arguments/sets of statements, whereas variables help us to say more about the system of logic in general.

In the previous section, we discussed the idea of a main connective, showing that every wff in our system can be reduced to a sentence that has just one of the five basic connectives. In this section we will pick that discussion back up, and use variables in the language to help discuss **logical forms**, the general shape that the sentences of the language can take.

Definition: A **sentence form** is one that contains only variables and sentence connectives.

We have seen sentence forms already throughout the course. The general forms for each of the basic connectives ('$p \vee q$', '$\neg p$', etc.) are all sentence forms. In addition, we have used sentence forms to describe other things, like the exclusive disjunction.

Sentence forms themselves are not sentences, because variables have no truth values – they are symbolic stand-ins, not fully realized statements. But if we take a sentence form and replace each variable with a wff using constants, we will always get a well-formed sentence. We can replace the variables with either

instance, if our sentence read "John will pass unless he *doesn't* study", then the negation of the statement after 'unless' would be that he *does* study. You need to keep track of your constants here so that you can get your symbolization correct.

atomic or compound sentences in each case if we want. This is called a **substitution instance**.

E.g. For particular statements A, B, C, and D, the sentence '$A \wedge B$' is a substitution instance of the sentence form '$p \wedge q$'. Similarly, '$(A \vee C) \wedge (B \wedge \neg D)$' is also a substitution instance of '$p \wedge q$'.

It is very important to realize that complex statements count as substitution instances for more than one sentence form. Let's continue using the statements A, B, C, and D, and the complex sentence '$(A \vee C) \wedge (B \wedge \neg D)$'.

As we have already seen, '$(A \vee C) \wedge (B \wedge \neg D)$' is a substitution instance of '$p \wedge q$', replacing the variable p with '$(A \vee C)$' and the variable q with '$(B \wedge \neg D)$'.

However, '$(A \vee C) \wedge (B \wedge \neg D)$' is *also* a substitution instance of the logical form '$(p \vee q) \wedge (r \wedge s)$', where we replace p with A, q with C, r with B, and s with $\neg D$.[17]

It is very important to get this stuff right. As we will see, being able to identify the various sentence forms present in a particular complex sentence is key to understanding the rules of logic. When we get to proofs in **Unit 2** this will all become much clearer, but we will use these techniques in just a bit when we begin computing complex truth truth tables.

Which of the sentence forms to use for a particular sentence at a given time with depend on the nature of the problem at hand, but here are some general rules that you can follow:

- Every wff, no matter how complex, has the logical form 'p'.
- Every sentence that has a negation as its main connective has the form $\neg p$.
- Every sentence that has a conjunction as its main connective has the form $p \wedge q$.
- Every sentence that has a disjunction as its main connective has the form $p \vee q$.
- Every sentence that has a conditional as its main connective has the form $p \implies q$.
- Every sentence that has a biconditional as its main connective has the form $p \iff q$.

[17] There is a further substitution instance, where the sentence form contains '$\neg s$' instead of 's'. In that case, the substitution would yield D in place of s.

As we have seen, more complex sentences sometimes have more sentence forms embedded within them. Once more, the key to becoming proficient at seeing all of the sentence forms present in a particular statement requires practice.

The specificity with which we express the sentence form will depend on our purposes, as we will see in the next sections pertaining to truth tables.

1.16 Computing Complex Truth Tables

So far, we have learned the truth tables for the basic logical connectives, and done a number of translations using them. This just scratches the surface of what is possible with our logical framework, however. The next step is to use the truth table method to analyze the truth conditions for more complex statements. This allows us to see when these more complex statements are true and false, based only on the truth values of their atomic parts. This has a number of applications, including being able to show validity, or when statements are logically equivalent to one another.

For our first example, we return to the exclusive sense of the disjunction. When I originally defined it, I gave its sentence form as '$(p \vee q) \wedge \neg(p \wedge q)$'. You may or may not have found that definition convincing when you first saw it, but now that we understand the syntax of well-formed sentences and the idea of sentence forms, we are in a position to prove that this definition captures the meaning of the exclusive disjunction.

The first thing that we do in setting up the truth table is check for the number of unique constants or variables. In this case there are two: p and q. This number dictates how many rows the truth table must have to give all of the truth conditions for the statement.

N.B. For a statement with n constants/variables, the truth table must have exactly 2^n lines.

That's a fancy way of saying that the number of lines in the truth table is multiplied by two every time there is an additional constant or variable. You can use this number to guarantee that your truth table is set up correctly. For one constant/variable, there will be two lines. For three, there will be eight lines, and so on. As with the truth tables for the basic connectives, we list all of the constants/variables on the left, with the complex statement heading the right-hand column. This is what the blank truth table would look like:

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

The next step is to fill in the truth values for the atomic statements. The easiest way to fill these columns out is to begin with the rightmost one first. In that column, simply alternate values of 'T' and 'F' until you fill all of the spaces, like this:

p	q	$(p \vee q) \wedge \neg(p \wedge q)$
	T	
	F	
	T	
	F	

Next, simply move one column to the left, and repeat the process, but this time alternate every two lines instead, i.e. 'T T F F'[18]:

p	q	$(p \vee q) \wedge \neg(p \wedge q)$
T	T	
T	F	
F	T	
F	F	

Finally, we're ready to start computing the truth values in the right-hand column of the truth table. The first step is to identify the main connective of the sentence. In this case, the main sentence connective is a conjunction (if you are still having trouble seeing this, revisit the earlier sections on the main connective and logical form). That means that we expect the final column of the truth table to be true when both conjuncts are true, and false otherwise.

However, each of the conjuncts themselves are complex statements, and thus in order to evaluate the truth value of the entire sentence, we first need to compute the truth values of the individual conjuncts. We will take it in two parts, starting with the left. I like to fill in truth tables one row at a time, so we will also go across the whole sentence before moving on to the next row.

The left-hand conjunct in this case is '$p \vee q$'. Fortunately for us, this is one of our basic connectives, and we already have the truth table for it memorized,

[18]If there were three variables instead, we would continue this process one more time, alternating after four lines: 'T T T T F F F F'.

so in the first row we can enter 'T' underneath the wedge, to indicate that the statement contained in the left-hand conjunct is true for that row of the table.

The right-hand conjunct reads '$\neg(p \wedge q)$', which is slightly more complicated than the left side, but not by much. It is a negation of a conjunction. The best way to determine its truth value is by working from the inside out. The innermost connective is the conjunction. On the first row of the truth table, the conjunction is true, since both of its conjuncts are true. Thus, under the carrot we enter 'T'. Now we deal with the negation, which is as easy as noticing that the statement being negated is true on this row, and so its negation must be false. We enter 'F' under the negation symbol.

All that's left to do now is to enter in the truth value of the main connective. It is a conjunction, with one true conjunct (on the left), and one false conjunct (on the right). Therefore, it must be false. We enter an 'F' under the main connective.

p	q	$(p \vee q) \wedge \neg(p \wedge q)$
T	T	T **F** F T
T	F	
F	T	
F	F	

That's the procedure for computing complicated truth tables: identify the main connective, and then evaluate the smaller statements in a manageable way. Take a minute to see if you can fill in the remainder of the table on your own. When you're done, come back and check your answer against the completed table.

All we have to do now is repeat the process for the remaining lines. This yields the following final truth table:

p	q	$(p \vee q) \wedge \neg(p \wedge q)$
T	T	T **F** F T
T	F	T **T** T F
F	T	T **T** T F
F	F	F **F** T F

Table 7: Truth Table for the Exclusive Disjunction

The last thing on the agenda is to take one final look at the table to make sure that it meets our expectations. Our aim was to give the meaning of the exclusive disjunction, an 'or' statement where exactly one of the disjuncts is true. The only lines where the right-hand column comes out true are lines two and three, where exactly one of the variables is true. That means that we got it exactly right.

Of course, this was just one example. The same method can be used to compute arbitrarily long and complex statements. You will get a good amount of practice computing truth tables, but as you can see it is a very mechanical procedure. Once you can read the statements well enough to identify the main connective and the scopes of the other connectives, you can complete such a truth table yourself. If you haven't already, now would be a good time to memorize the truth tables for the basic connectives. It will dramatically speed up your work.

1.17 Contingent Sentences, Tautologies, and Contradictions

In this section we will learn about the three logical possibilities for the truth-value assignments of sentences. Every sentence in the language has exactly one of these three features. Furthermore, computing the truth table for a particular sentence will determine which of these features holds for that sentence.

1.17.1 Contingency

Definition: A sentence form is **contingent** if there is at least one truth-value assignment of its atomic components that makes the sentence true, and at least one truth-value assignment that makes the sentence false.

Logical contingency is a fancy way of saying 'it's sometimes true, and sometimes false'. For an example, we need look no further than the exclusive disjunction that we just covered. Any substitution instance of the exclusive disjunction is contingently true – it depends on whether exactly one of its disjuncts is true.

N.B. You can tell that a sentence form is logically contingent if the final column of its truth table has at least one row that has the value 'T' and one that has the value 'F'.

1.17.2 Tautologies

Consider the sentence "Either Trinidad and Tobago gained independence from Great Britain on August 31st, 1962, or Trinidad and Tobago did not gain independence from Great Britain on August 31st, 1962". Is this sentence true or false? You probably don't know much about the history of a small country in the Caribbean, but despite that I bet that you knew this sentence was true. Why? Well, because it had to be! If that's when Trinidad and Tobago gained independence, then the sentence is true, and if it isn't when they gained independence, then the sentence is also true. We call such a statement a **tautology**.

> **Definition:** A sentence form is a **tautology** if every truth-value assignment of its atomic components makes the sentence true. A tautology is true in virtue of its logical form.

The example that we just saw is a substitution instance of the sentence form '$p \vee \neg p$'. Any sentence with that form will also be a tautology! That means that there are an infinite number of tautologies just based on this sentence form alone.

Tautologies aren't always easy to spot. Consider the sentence form '$p \implies (q \implies p)$'. You might not be able to think of any substitution instances for it, because it looks pretty odd. But if we compute its truth table, we get[19]:

p	q	$p \implies (q \implies p)$
T	T	T
T	F	T
F	T	T
F	F	T

No matter what the truth-values are for p and q, this sentence form is always true. Therefore, it is a tautological sentence form, and any of its substitution instances will be tautologies.

Tautologies are interesting because they are not always easy to spot. If we could identify tautologies more easily, we wouldn't have as much work to do in this class, since we could then identify valid arguments with no issue[20]. Tautologies also have a funny place in the English language, because they are used surprisingly often. For instance, have you ever heard someone say "We'll get there when we get there" or "It is what it is'? Those are both examples of tautologies that are used in English. Funny enough, they are sometimes used in order to sound deep or profound, but what could possibly be profound about something that's logically true? It's not like it tells us anything interesting!

1.17.3 Contradictions

The last possibility is that of logical contradiction. Where a tautology is always true, a logical contradiction is always false.

[19] I will leave it to you to fill in the additional truth-value assignments for the right-hand column if you feel the need to verify that the truth table is correct

[20] I don't expect you to see why this is the case as yet!

Definition: A sentence form is a **contradiction** if every truth-value assignment of its atomic components makes the sentence false. A contradiction is false in virtue of its logical form.

A simple example of a contradiction would be any sentence of the form '$p \wedge \neg p$'. For example, "It is snowing, and it isn't snowing" ; "Vanilla is a popular flavor of ice-cream, and vanilla is not a popular flavor of ice cream".

Tautologies and contradictions share an interesting relationship. If you negate a tautology, you get a contradiction[21], and vice versa. And just like tautologies, there are some more complex contradictions that are harder to spot. Again, if it were easy for us to identify all of the logical contradictions, then our goal of identifying valid arguments would be trivial!

The notion of a contradiction is extremely important in the study of logic. In **Unit 2**, we will learn an extremely powerful method for proving deductive validity that is predicated entirely on contradictions.

We have now covered all of the logical possibilities for individual sentence forms: an individual sentence form must be contingent, a tautology, or a contradiction. Next, we will consider two features of *sets* of statements, **consistency** and **logical equivalence**.

1.18 Consistency

Definition: A set of statements is **logically consistent** if it is possible for every statement in the set to be true at the same time. Otherwise, the set is **inconsistent**.

It is easy to gets contingency and consistency mixed up. The key is to remember that being contingent is a feature of a single sentence form, while consistency is a feature of sets of statements. It takes at least two statements to check for **consistency**.

As an easy example, the statements '$A \vee B$' and '$A \wedge B$' are logically consistent, since it is possible for them to both be true at the same time (when both A and B are true).

[21] That means that the sentence form '$\neg(p \implies (q \implies p))$' is contradictory, for example.

1.19 Logical Equivalence

Definition: Two sentences are **logically equivalent** if they share the same truth values in all cases. Every line of their respective truth tables are identical to one another.

Logical equivalency is immensely important. Recall when we first began outlining the logical connectives we specified that the truth-value assignments for the truth tables effectively captured the logical meaning of the statements. Therefore, if two statements are logically equivalent to one another they have the same exact meaning. Whenever it would be appropriate to assert or conclude one, it would be appropriate to assert or conclude the other as well.

We have already run across an example of logically equivalent statements. During our discussion of English 'unless' sentences, I noted that such sentences could be translated in two different ways. For a sentence 'A, unless B' with some particular statements A and B, both '$\neg B \implies A$' and '$\neg A \implies B$' were accurate translations. Now that we have a definition for logical equivalency, we can use the truth table method to prove that these two statements are, in fact, equivalent to one another.

We will do this by showing that the sentence forms '$p \implies q$' and '$\neg q \implies \neg p$' are logically equivalent. In order to do that, let's first show what is probably the simplest case of logical equivalency: that 'p' is logically equivalent to '$\neg\neg p$'.

This would be fairly easy to do using the truth table method, but it's just as easy to informally justify it[22]. Suppose p has the value 'T'. By the definition of negation, $\neg p$ has the value 'F'. Then, $\neg\neg p$, which is the negation of $\neg p$ must have the value 'T'. The same reasoning works if p has the value 'F'. Therefore, no matter what truth value p has, $\neg\neg p$ has the same value, so by definition they are logically equivalent.

Now onto the truth table for the two conditional forms. After constructing the truth table, we get the following result:

If you are struggling to see the equivalency without filling in the truth-values for the negations and verifying the final truth-values for the conditional in the right-hand column, that's OK. It takes a bit of practice to build up the skill to hold the truth-values in your head. If it would be helpful for you, write out the table yourself and verify this result.

[22] It's also good practice to try to talk through this kind of reasoning, so that you are constantly deepening your understanding and independently confirming what the truth table shows.

p	q	$p \implies q$	$\neg q \implies \neg p$
T	T	T	T
T	F	F	F
F	T	T	T
F	F	T	T

Table 8: Proof of Equivalency Between a Conditional and its Contrapositive

The truth table shows definitively what we had already suspected: these two sentence forms are logically equivalent! This equivalency is extremely helpful for us in solving proofs, as we will see in **Unit 2**.

You might have noticed something about the definition of logical equivalency as it pertains to our basic logical connectives. When we discussed the biconditional, we saw that it aimed to capture the idea that two statements always shared the same truth value. But that's the definition of logical equivalency! What this means is that we can use the biconditional as a way to express that two statements are logically equivalent to one another.

N.B. The biconditional between two logically equivalent sentence forms is a tautology.

In this case, we have the sentence form $(p \implies q) \iff (\neg q \implies \neg p)$. This notion – that the biconditional between logically equivalent sentence forms is always true – is another fundamentally important bit of our logical machinery. As you will see when we get to proofs, this one fact will allow us to dramatically increase the efficiency with which we perform proofs.

1.20 Using Truth Tables as a Test for Consistency/Validity

By now, we've gained a lot of practice when it comes to computing more complex truth tables and doing translations from English. At the same time, our theoretical understanding of the system of sentential logic has improved, owing to our discussions of consistency, tautologies, logical equivalence, and so on. All that's left for us to do now is put everything together towards our final goal: deciding when an argument is or is not deductively valid.

1.20.1 The Truth Table test for Consistency

In order to achieve this goal, we need to understand how to use truth tables to test for both consistency and validity. We start with consistency. Consider the following statements:

1. $A \implies \neg B$
2. $B \vee C$
3. $\neg A$

Are 1-3 consistent? That is, is it possible for them to all be true at the same time? We can use a truth table to find out. This is what we end up with:

A	B	C	$A \implies \neg B$	$B \vee C$	$\neg A$
T	T	T	F	T	F
T	T	F	F	T	F
T	F	T	T	T	F
T	F	F	T	F	F
F	T	T	**T**	**T**	**T**
F	T	F	**T**	**T**	**T**
F	F	T	**T**	**T**	**T**
F	F	F	T	F	T

I have highlighted the pertinent lines. What they tell us is that the set is indeed consistent, since there are truth value assignments for the constants that make all three statements true at the same time. This is all it takes to test a set of statements for consistency: so long as all of the statements are true on at least one line, the whole set is consistent.

1.20.2 The Truth Table Test for Validity

Now we can move on to analyzing arguments for deductive validity. We will use the same truth table method as before. Consider the following argument:

1. $\neg A \vee B$
2. $\neg B \vee C$
3. $\neg C$

$\therefore$ 4. $\neg A$

Now, we might be tempted to start constructing the truth table right away. However, this would technically be a mistake. Remember, deductive validity is not a feature of particular arguments, but rather of argument *forms*. Therefore, it's only appropriate for us to check for validity when we've swapped out the particular statements for variables instead.

However, there are various ways to substitute particular sentences for general forms, as we saw earlier on in this unit. For example, we saw that a sentence like '$\neg A \vee B$' has the logical form '$p \vee q$', but that it also has the logical forms

'$\neg p \vee q$' and 'p', depending on what level of specificity a particular situation requires.

This is exactly the sort of situation where we need to be careful about which forms we choose. For instance, it would be wrong of us to represent the form of (1) as 'p', because it contains particular statements that are also present in later parts of the argument. Thus, if we pick this form we lose some of the essential information of the argument. What we need to do instead is choose the broadest or most expanded form possible, which basically amounts to replacing each constant with a variable, **ensuring that the same variable is used for every instance of that constant occurring in the set**.

In this example, then, if we replace the constant A in (1) with the variable p, then we must also replace the occurrence of A in the conclusion with p. The same thing follows for each of the other constants.[23] This yields the follow argument form:

1. $\neg p \vee q$
2. $\neg q \vee r$
3. $\neg r$

$\therefore$ 4. $\neg p$

Now that we have successfully put the argument into its general form, we are ready to construct the truth table. Before you look at the truth table, see if you can reason through the argument form and determine whether or not it is valid. Remember, an argument is deductively valid if, and only if, it is impossible for all of the premises (1-3 here) to be true while the conclusion (4) is false. In the context of the truth table, this means that the argument form is valid if every row where all of the premises are true is also a row where the conclusion is true. Without further ado, let's look at the truth table.

[23] Many of you will wonder why the preceding two paragraphs were necessary at all. After all, isn't the most expanded form the most obvious way to represent the sentence form in the first place? That might well be true, but it is important to be precise, so it felt like a mistake for me to assume this, given the discussion of logical form that we have already had.

p	q	r	$\neg p \vee q$	$\neg q \vee r$	$\neg r$	$\neg p$
T	T	T	T	T	F	F
T	T	F	T	F	T	F
T	F	T	F	T	F	F
T	F	F	F	T	T	F
F	T	T	T	T	F	T
F	T	F	T	F	F	T
F	F	T	T	T	F	T
F	F	F	**T**	**T**	**T**	**T**

As usual, the key line is highlighted in bold. The only line where all of the premises are true is row 8, and on that row the conclusion is also true. This argument form, then, must be valid. It is impossible for a substitution instance of this form to have true premises and a false conclusion. What's more, the procedure that we just employed works for any argument form. The truth table method provides a universal test for validity.

In the section on tautologies, I mentioned that the notion of a tautology is intimately related to our inquiry into valid argument forms. Now that we see have seen a truth table test for validity, I can better explain the connection. Notice that an argument form is *invalid* if it is possible for all of its premises to be true but its conclusion false. Next, recall the truth table for the material conditional. It is false on only one row, when the antecedent is *true* but the consequent is *false*. Last, recall that a conjunction is only true when all of its conjuncts are true. If we combine these ingredients, we can demonstrate the relationship between tautologies and valid argument forms nicely.

Consider a valid argument consisting of premises $\{P_1, P_2, ...P_n\}$[24] and conclusion C.

Now consider the conjunction of all of the premises, $(P_1 \wedge P_2) \wedge (P_3 \wedge P_4) \wedge ... \wedge P_n$. Because this is a conjunction, it is only true when all of the premises are true at the same time.

Finally, consider the material conditional obtained by taking $((P_1 \wedge P_2) \wedge (P_3 \wedge P_4) \wedge ... \wedge P_n)$ as its antecedent and C as its consequent. This yields

$((P_1 \wedge P_2) \wedge (P_3 \wedge P_4) \wedge ... \wedge P_n) \implies C.$

[24] This is notation common in mathematics. Don't be alarmed by it. This basically says "here is a set of premises". We use the subscripts, the ellipsis, and the subscript 'n' at the end to say that this is an arbitrarily long list, because we are trying to show a result that holds for all cases.

However, since we started off with the assumption that the set of premises along with the conclusion constituted a valid argument form, whenever all of the premises are true, the conclusion must also be true. Therefore, when the premises are all true, the entire conditional is true.

Furthermore, if any of the premises is false, the whole conjunction – and thus, the antecedent – will be false. Whenever a material conditional has a false antecedent, the entire conditional is true. As such, this conditional will always be true no matter the truth values of the individual statements. By definition, then, this conditional is a tautology. ■[25]

This result means that any valid argument form has a corresponding tautology associated with it, which is why I said earlier that if we knew all of the tautologies it would make our job of finding valid arguments quite easy!

1.20.3 Using Truth Tables to Develop Counterexamples

We have just seen the truth table for a valid argument. But what happens when our argument form is invalid instead? As you might expect, this would imply that all of the premises are true, but the conclusion false on at least one row of the truth table. In this section, we will see how we can use this fact to help construct counterexamples for invalid logical forms.[26]

The following argument form is invalid. Instances of this kind of bad argument are so common it even has a name, 'affirming the consequent'.

1. $p \implies q$
2. q

$\therefore$ 3. p

Luckily for us, this argument uses all of the same components as the truth table for the material conditional. We need to keep straight which rows are premises and which is the conclusion, though.

[25] This black square, sometimes called a tombstone, is something that mathematicians and logicians place at the end of a proof to indicate that they have shown the result that they wanted to. More familiar to some people are the letters QED, which indicate the same thing. You might think it strange that I placed this here, but this is a successful proof of the fact that every valid argument form has a corresponding tautology! We won't be doing many of these 'meta' proofs in this course, but I thought I'd throw this one in because it's interesting and not so hard to follow. Pretty neat, huh?

[26] We've already seen one example of a counterexample, in **Section 1.3.2**. If we knew how to translate the premises of those arguments into our logical notation, we could use that as our example here, but for that we need to wait until **Unit 3**.

p	q	$p \implies q$
T	T	T
T	F	F
F	**T**	**T**
F	F	T

As we can see on row 3, both of the premises are true, while the conclusion, p, is false. This confirms that the argument form above is invalid. But we can glean even more information from the truth table, because we can see the specific truth values that each variable has for this line. What that means is that we can create a truth-value assignment to use as the basis for a counterexample.

Reading off the truth values on row 3, if we can come up with a conditional with a false antecedent, but a true consequent, we can give a counterexample in English that shows that any argument with this form is invalid. For this example, we will let $p =$ "Elon Musk is the CEO of General Motors" (which is false, as per row 3), and $q =$ "Elon Musk is the CEO of a car company" (which is true).

1. If Elon Musk is the CEO of General Motors, then Elon Musk is the CEO of a car company.
2. Elon Musk is the CEO of a car company.

$\therefore$ 3. Elon Musk is the CEO of General Motors.

And there you have it, an English counterexample to this invalid argument form. Of course, generating the English counterexample takes a bit of creativity and practice, but the truth table is a helpful guide, since it gives a model for which atomic statements need to have which truth-values in the counterexample.

1.20.4 Summary

This brings us to the end of **Unit 1**, and so concludes our introduction to all of the technical machinery of sentential logic. Before we move on, make sure you feel comfortable with

- The concept of deductive validity.
- Truth-values and the axioms of sentential logic.
- The basic connectives and their corresponding truth tables.
- Identifying a sentence's main connective and logical forms.
- Translating complex sentences from English into logical notation.
- Using the truth table method to test statements for consistency, validity, etc.

In **Unit 2**, we move on to proofs. We will use all that we learned in this unit to learn the system of **natural deduction**, a method that allows us to show validity and logical equivalence in a much more streamlined fashion than the truth table method.

2 Unit 2: Proofs and Natural Deduction

2.1 Introduction

At the end of **Unit 1**, we learned about the truth table method, and how it could be used to determine logical validity/invalidity. We saw two nice features of using truth tables: (1) it is a purely mechanical process, with no guesswork, and (2) it is an infallible procedure. Furthermore, we saw that truth tables could also help us create models for counterexamples to invalid arguments.

However, truth tables are also severely limited once arguments become more complex. Imagine an argument form that contains seven different variables. In order to apply the truth table method, we would need to construct a truth table of 2^7 – or one hundred and twenty-eight – lines! It already takes long enough to compute a truth table with just four or eight lines. In fact, given enough variables, even a computer working around the clock would take a substantial amount of time to compute a truth table.

For this reason, logicians have developed an alternative method for demonstrating an argument's validity, called the **method of proof** or **natural deduction**. In this unit, we will be learning all about this method, how it works, and how to use it to prove the validity of different argument forms of varying complexity.

2.2 What's in a Proof?

As usual, we'll start off with a formal definition of a proof. Don't worry about not understanding it fully, because we're going to go through it one step at a time and I'll give a full explanation.

Definition: A **proof of** A – for some wff A – is a set of numbered statements $\{P_1, P_2, ..., A\}$ such that each statement is either:

1. A premise of the argument, **put forward** or **given as true**, or
2. derived from one or more premises by a **valid rule of inference**, ending with the conclusion A.

The easiest way to understand the definition is to go through a specific example of a valid argument form. Here's a very simple one that we encountered in **Unit 1**.

1. $p \implies q$
2. p

$\therefore$ 3. q

This isn't a proof as yet. So far, all we have are the collection of premises and the conclusion. In order to transform this into a proof, we would need to add in the **rules of inference** that allow us to validly arrive at the conclusion based on these premises.

> **Definition:** A **rule of inference** is a function that takes sentence forms as inputs, and gives a different sentence form as an output, as specified by the particular rule in question.

"Oh boy!", you may be thinking, "One totally unintelligible definition after another!". Luckily, these formal definitions only look scary – they're actually pretty intuitive to grasp. Let's go back to our examples – or more specifically, just the premises – to see what I mean.

1*. $p \implies q$

2*. p

We've isolated just two lines of our original argument form. Let's take a good, hard look at them. 1* is a conditional, and 2* is the antecedent of that conditional. Now here's the truth table for the material conditional that we're already very familiar with:

p	q	$p \implies q$
T	T	T
T	F	F
F	T	T
F	F	T

As we can see from the truth table, if we know that a conditional is true, and that its antecedent is also true, then the only possibility is that its conclusion is true. In other words, this argument form is valid, as per the truth table.

The idea, then, is that we can use this very simple valid argument form to create a **rule of inference**. The way that the rule works is that whenever we have a premise in our argument that has the form '$p \implies q$', and another premise that has the form 'p', we are **allowed to introduce a new line that says**

'q'. The order that the lines occur is irrelevant – all that matters is that both of the lines exist **somewhere** in the proof.

We call these *rules* of inference because they tell us when we're allowed to use the existing lines of a proof to generate new lines. This particular rule is called *modus ponens*, which is a Latin phrase that I think means something about affirming. For the sake of not learning any Latin, we'll just call the rule '**MP**'. We write it out like this:

Rule MP:

$$\begin{array}{l} 1.\ p \implies q \\ 2.\ p \\ \hline 3.\ q \end{array}$$

The way we read this is the same as what's written in the previous paragraph: "Whenever you have lines of a proof with the same forms as 1. and 2., you are permitted to make a new line with the same form as 3.". Now we can go back to the original argument, and add in the rule that we used to go from the premises to the conclusion.

1.	$p \implies q$	
2.	p	
$\therefore$ 3.	q	[From lines 1,2 **MP**]

Here, we've written the argument over, but this time we have cited the relevant lines, and the relevant rule that we used to obtain the conclusion from the premises. This was the final ingredient that was missing, and so what we now have is a **proof** of q **from** the premises $p \implies q$ and p.

To summarize, a proof consists of a set of premises, along with additional lines that are derived from those premises using the rules of inference, ending finally in the conclusion.

N.B. Every time you introduce a new line into a proof using a rule of inference, you must cite the lines and the rule that allowed you to do so.

The reason that we stress the citations for every new line is to ensure that our proof follows all of the rules. It's a built-in way of checking our work as we go.

2.2.1 A Slightly More Complicated Example

At the moment we only have one rule, **MP**. Before moving on to the other rules and adding in more of the theoretical framework, let's do a slightly more difficult example. We're going to prove that the following argument is valid:

1. A
2. $A \implies \neg C$
3. $\neg C \implies B$

$\therefore$ 4. B

We start every proof by simply restating all of the premises. These are the statements that are given to us as true that we are going to use to show that the conclusion must also be true.

1. A
2. $A \implies \neg C$
3. $\neg C \implies B$

We only know one inference rule. That should give us a pretty big hint about what to do next, so let's look through the proof for lines that have the same forms as those in the rule. Immediately, our eyes should be drawn to lines 1 and 2. These lines have the correct sentence forms to fit within our rule, so we can use **MP** to generate a new line as per the rule. This gives us:

1. A
2. $A \implies \neg C$
3. $\neg C \implies B$
4. $\neg C$ [1,2 **MP**]

We've now introduced a new line (4) into our proof, using lines 1, 2, and our inference rule **MP**. But this isn't all that we can do! Now that we have line 4 as part of the proof, we have *another* instance of two lines that have the same forms as those required to use **MP**. In this case, it's lines 3 and 4. We continue the proof like so:

1. A
2. $A \implies \neg C$
3. $\neg C \implies B$
4. $\neg C$ [1,2 **MP**]
5. B [3,4 **MP**]

And look at that! This second application of **MP** gets us to the conclusion. Let's do a quick recap to make sure this is a good proof. We'll check each of the lines of our proof against the definition given at the beginning of the section.

- Is line 1 either a premise or derived from one or more premises by a valid application of a rule? **A: Yes, it is a premise.**

- Is line 2 either a premise or derived from one or more premises by a valid application of a rule? **A: Yes, it is a premise.**

- Is line 3 either a premise or derived from one or more premises by a valid application of a rule? **A: Yes, it is a premise.**

- Is line 4 either a premise or derived from one or more premises by a valid application of a rule? **A: Yes, it is derived from lines 1 and 2 using a valid application of rule MP**.

- Is line 5 either a premise or derived from one or more premises by a valid application of a rule? **A: Yes, it is derived from lines 3 and 4 using a valid application of rule MP**.

Since line 5 is also the conclusion, and every line of the proof was obtained in a valid way, this is definitely a proof of B from the premises A, $A \implies \neg C$, and $\neg C \implies B$.

2.3 The Limits of Proofs and Some Important Reminders

Natural deduction is an amazing system, allowing us to show validity much more seamlessly than truth tables (once we are familiar with all the rules and get come practice in, that is). However, there are some limitations that need to be kept in mind when doing proofs.

- Proofs can only show **validity**. There is no way to use natural deduction to prove that an argument is **invalid**. This is another reason why it is so crucial to justify every step of your proofs.

- The rules only apply to **whole lines**, and never to parts of a line.

- The rules of inference require you to follow the sentence forms exactly.

It will be helpful if we go over the last two bullet points before moving on to other rules.

E.g. 1 Take a look at the following premises:

1. $(A \implies B) \implies C$
2. A

The question is, "Can we use **MP** on lines 1 and 2?". Think about your answer for a minute before moving on to the explanation.

The answer is 'no'. This is because those two lines don't have the correct forms to be used with the rule. The rules only apply to entire sentences. In order to check to see whether a rule can be validly applied, you must convert the entire sentence into its variable form, focusing on the main connective, just like we saw in **Unit 1**.

In this case, the main connective of line 1 is the conditional on the right. This would yield the logical form '$p \implies q$'. But now that we have substituted the complex expression $(A \implies B)$ for the variable p, we can no longer use p as a variable substitution for the atomic sentence A. Therefore, if we change both of these premises into their general forms as given by their main connectives, we get

1*. $p \implies q$

2*. r

which, as you can now see much more clearly, does not have the right form to use **MP**.

The easiest way to remember these rules is to focus on the main connective of each sentence, and treat the statements on either side of the main connective as a shape. When you're trying to use a rule, all you are doing is matching the shapes up in the correct way. Here's one more example to see what I mean.

1**. $(A \implies (B \vee C)) \implies D$

2**. $A \implies (B \vee C)$

In the above example, we still have a conditional in line 1**. But this time the antecedent is exactly the same as what's given in line 2**. Therefore we *can* use **MP** in this case to get a new line that just says 'D'.

In summary, here is a breakdown of what we did. First, we found the main connective in 1** – the conditional on the right. Then, we checked to see that 2** has the exact same shape as the antecedent of the main conditional in 1**, and we found that they do (since they both say exactly '$A \implies (B \vee C)$'. That allows us to use **MP**.

Applying these rules is like playing a complicated version of the children's game of matching differently shaped pegs with holes in a block of wood. First, you identify the main connective (Is this a round hole, a square hole, etc.?). Then you try to match the pieces together in the right way (Does this peg fit with that hole?). Getting good at it is just a matter of practice.

2.4 The Rules of Inference

Next up is laying out all of the rules of inference. It turns out that we have some freedom to move when it comes to the rules that we pick. There are two criteria that we would like to ensure. First, we want to make sure that everything that we can prove is actually valid (so we never accidentally use a correct proof to prove something that is invalid). Second, we want to make sure that if something is valid, then there will be a way for us to prove that it's valid. These criteria are called **soundness** and **completeness** respectively. More advanced logic courses go so far as proving these features (we will just assume them).

Fortunately, this leaves us with a decent amount of latitude when it comes to the rules. But there are other reasons to favor one ruleset over another. For example, we might want what's called an 'elegant' set of rules – one that is very minimal but nonetheless powerful. Sometimes, mathematicians and philosophers calls such a set **parsimonious**.

On the other hand, we also want the rules to be convenient and reasonably easy to use. As such, we may want to include more rules than are strictly necessary to make proofs less difficult. The result is a set that strikes a decent balance: not extremely minimalist, but also not needlessly bloated.

As the course goes on, we'll learn how to use natural deduction to derive new rules, in case we ever feel like adding some more will make our lives even easier. I will also provide every student with a separate document that has all the rules on it for easy referencing, once we've gone through the explanations for each.

2.4.1 Inference Rules for Negations

We have two rules for negations, but really they are just variants of one another.

Rule DN1:

1. $\neg\neg p$

---

2. p

Rule DN2:

$$\begin{array}{l} 1.\ p \\ \hline 2.\ \neg\neg p \end{array}$$

These rules are pretty easy for us to justify. As we know, negations always have the opposite truth value to what they are negations of. Adding another negation, then, gives us the same truth value as the original statement. So we can always conclude a statement from its **double negation**[27], and vice versa.

N.B. In a proof, one can always generate a new line $\neg\neg p$ from a line p, and a line p from a line $\neg\neg p$.

Here are a couple examples of the rules in action.

1. $\neg\neg\neg A$
2. $\neg A$ [1, **DN1**]

In this example, notice that the first line actually has three negations. However, because we can represent the general form of that statement as $\neg\neg p$, where p is a variable substituted for '$\neg A$', it has the correct form to use the rule.

1. $A \vee \neg B$
2. $\neg\neg(A \vee \neg B)$ [1, **DN2**]

Here, we apply the rule **DN2** to the first line. Notice that we have to add the negations to the entire sentence, and so the first step is to enclose the sentence in a pair of parentheses, to ensure that the negations have the correct scope.

2.4.2 Inference Rules for Conjunctions

We have two rules for conjunctions, though it is helpful to clarify that the rules can be implemented in logically equivalent – but syntactically different – ways.

Rule Simp:

$$\begin{array}{l} 1.\ p \wedge q \\ \hline 2.\ p \end{array}$$

[27] That's why the rules are called 'DN'.

This rule (called "**Simplification**", or "**Simp**" for short) says that if you have a conjunction, you can generate a new line that is just one of the conjuncts. This follows from the fact that a conjunction can only be true if both conjuncts are true. The important thing to note is that one can use the exact same rule in the following way:

$$\begin{array}{l} 1.\ p \wedge q \\ \hline 2.\ q \end{array}$$

The rule allows us to generate a new line with *either* conjunct, not just the one on the left. This should be intuitive, but not every rule works this way, so clarifying is important.

Next we have a rule (called "**Conjunction**") for introducing new conjunctions into our proofs. As you might expect, we can combine any two existing lines of a proof into a conjunction with each of those lines as conjuncts.

Rule Conj:

$$\begin{array}{l} 1.\ p \\ 2.\ q \\ \hline 3.\ p \wedge q \end{array}$$

Just as with **Simp**, the specific order of the conjunction doesn't matter, so if we wanted we could also add the conjunction as '$q \wedge p$'. Here is an example applying the rules in several different ways.

1. $(A \wedge B) \wedge C$	[**Pr**]
2. $D \vee E$	[**Pr**]
3. $(A \wedge B)$	[1, **Simp.**]
4. A	[3, **Simp.**]
5. $(D \vee E) \wedge A$	[2,4 **Conj.**]

In this example, we have two premises on lines 1 and 2. On line 3, we use **Simp** on line 1 to generate a new line that just features the first conjunct of that line. However, line 3 is itself a conjunction, so it is now possible to use **Simp** again to generate line 4. Finally, we can use line 2 and line 4 to generate line 5, which is the conjunction of the other two.

2.4.3 Inference Rules for Disjunctions

Moving on to disjunctions, things start to get a bit less intuitive. The rules should still make sense after a little discussion, but you might need to stop and read them over once or twice. As usual, all of these rules can be verified as valid rules of inference by the truth table method.

Rule DS:

1. $p \vee q$
2. $\neg q$

---

3. p

This rule, which is called "**Disjunctive Syllogism**" tells us that whenever we have a disjunction, and another line that is the negation of one of the disjuncts, we can generate a new line consisting of the other disjunct. Such a rule should be intuitively justified to us, since we know that the disjunction means "one, the other, or both". Therefore, if we know that the whole disjunction is true, but one of the disjuncts is false, then the other disjunct must be true. Just as with **Simp**, the rule has two variants. The other is

1. $p \vee q$
2. $\neg p$

---

3. q

This rule is very important because it allows us to narrow down the possibilities presented by a disjunction when we find out that one of the disjuncts is false. Any time you see a disjunction in a proof, always be on the look out for the negation of one of the disjuncts.

Continuing to the next rule, we have one that surprises a lot of students.

Rule Add:

1. p

---

2. $p \vee q$

Addition (**Add**) is a somewhat surprising rule. It tells us that we can take any line of a proof and generate a new line by disjoining the original with *any statement whatsoever*. The reason this rule usually surprises people is that it is *semantically counter-intuitive.* If you know, for instance, that Bob likes chocolate cake, it would seem very strange to say that "Bob likes chocolate cake *or* he likes cheesecake". That would violate a norm of communication, because you would be implying that you didn't know which specific kind of cake Bob liked. In general we tend to assert the strongest or most certain version of what we believe. However, we're only interested in the logic of the statements and their truth values. A disjunction with a true disjunct is no less *true* than the true disjunct alone. Hence, we have this rule of inference. It's a hard one to remember because it's so foreign to our usual way of speaking, so keep it in mind!

There is one other rule that touches on disjunctions. It's something of a combination rule, since it also features the material conditional.

Rule CD

1. $p \vee q$
2. $p \implies r$
3. $q \implies s$

---

4. $r \vee s$

Constructive Dilemma is related to **MP** that we saw at the beginning of the chapter. In a way, it's like a disjunctive version of it. We can intuitively justify the rule with a kind of informal story.

"I know that either p or q is true (or both are). I also know that whenever p is true, r is true (line 2), and that whenever q is true, s is true (line 3). But I don't know *which* of the statements is actually true! The best I can do, then, is conclude that *at least one* of r or s must be true, which is just the same as saying that $(r \vee s)$ is true."

On **Problem Set 3** we saw a special case of **CD** where both conditionals had the same consequent.

1. $p \vee q$
2. $p \implies r$
3. $q \implies r$

$\therefore$ 4. r

As we proved via the truth table method, this argument is valid. Now we have the ability to prove it using natural deduction. All we need to do is apply **CD** to the premises.

1. $p \vee q$ [**Pr**]
2. $p \implies r$ [**Pr**]
3. $q \implies r$ [**Pr**]
4. $r \vee r$ [1,2,3 **CD**]

This is basically the whole proof, once we realize that '$r \vee r$' is identical to r.[28] In that case, this is already a good proof of the validity of this argument form. Later on in this unit, we will learn an additional proof technique that allows us to use this special case of **CD** to great effect.

2.4.4 Inference Rules for the Material Conditional

The inference rules for conditionals are some of the most useful, showing up in a large percentage of common proofs. We've already gone over **MP**, so we can avoid rehashing it. Instead, let's move on to the next inference rule on the list, *modus tollens*.

Rule MT

1. $p \implies q$
2. $\neg q$

---

3. $\neg p$

MT (still avoiding learning any Latin) is an analogue of **MP**, in the sense that they both come from the truth table for the material conditional. There is only one possibile truth-value for the antecedent of a true conditional with a false consequent – it must be false.[29] Therefore, whenever we have a conditional and the negation of its consequent in a proof, we can conclude that the antecedent must be false, and generate a new line accordingly. We can see this in action in the following example.

1. $A \implies (B \wedge C)$ [**Pr**]
2. $\neg(B \wedge C)$ [**Pr**]
3. $\neg A$ [1,2 **MT**]

[28] N.B. This notion of identity is actually stronger than our previous notion of logical equivalence. We don't have the time or space to stop and talk about it now (usually it's saved for an intermediate class), so you may just have to take my word for it.

[29] Go back to the truth table for the conditional to verify this fact.

MT is a very powerful rule, but it can be difficult to apply some times if you're not careful. In the following example, for instance, it's possible for us to apply **MT**, but only after a few crafty steps.

1. $A \implies \neg B$ [**Pr**]
2. B [**Pr**]

It is possible to use **MT** on these two lines, but since we are being strict about our sentence forms matching exactly, we cannot as yet employ the rule. That's because 'B' doesn't have the form of being the negation of '$\neg B$'. Luckily, we have a rule that allows us to generate the correct form: **DN2**.

3. $\neg\neg B$ [2, **DN2**]

Now that we've done this, we do have the correct forms in order to use **MT**, thus yielding

4. $\neg A$ [1,3 **MT**]

This is a common trick in proofs, so try to make a note of it for later when we start doing problems.

The last rule for conditionals is called **Hypothetical Syllogism** (**HS**). It is a rule that some do without since, as we will see, the same effect can be achieved by successive applications of **MP**. However, it is a nice tool to have in our kit for speeding up our solving. Without further ado,

Rule HS

1. $p \implies q$
2. $q \implies r$

---

3. $p \implies r$

The easiest way to think of applying this rule is that we're 'gluing' two conditionals together end-to-end. Let's think the justification through. Let's assume that p is true (that's the 'hypothetical' part of 'hypothetical syllogism'). If p is true, we can use **MP** to show that q is true. But now that we know that q is true, we can use **MP** again to get that r is true. So it follows that whenever p is true, r is true as well. Hence, $p \implies r$.[30]

HS is a very convenient rule to have, since it lets us skip clogging up our proofs with repeated applications of **MP** in some cases. In other cases, the thing

[30] Once we have some practice doing longer proofs under our belt, we will learn a generalized way to apply this reasoning to our proofs, allowing us to generate new conditionals in our proofs when needed.

we want to prove is itself a conditional, in which case this rule is even more helpful.

2.4.5 Rules of Inference for Biconditionals

This is the final section before we start doing proofs in earnest. There are two rules for biconditionals, one for decomposing them into conditionals, and one for building them up. In this way, the rules for biconditionals are very similar to the rules for conjunctions.

Rule Bisimp

$$\begin{array}{l} 1.\ p \iff q \\ \hline 2.\ p \implies q \end{array}$$

As with **Simp**, **Biconditional Simplification** can be used to decompose a biconditional into two conditonals. Recall in **Unit 1** that we said that $A \iff B$ is logically equivalent to $(A \implies B) \wedge (B \implies A)$. As such we can generate either of these two conditionals whenever we have a line with a biconditional. The other variant of the rule looks like

$$\begin{array}{l} 1.\ p \iff q \\ \hline 2.\ q \implies p \end{array}$$

The final rule of our initial set is the reverse: **Equivalence** ('**Equiv**') allows us to take two conditionals with the appropriate sentence forms and generates a new line with a biconditional.

Rule Equiv

$$\begin{array}{l} 1.\ p \implies q \\ 2.\ q \implies p \\ \hline 3.\ p \iff q \end{array}$$

Of course, since we could have the two conditionals in either order, we could also generate the line '$q \iff p$'. Since these rules function just like the rules for conjunctions, let's skip an example and dive right into some step-by-step examples of proofs and how to go about solving them on your own.

2.5 Solving Proofs: A Step-by-step Walk-through

At this point, we have all of our initial rules for solving proofs. You might grasp all of the justifications for the rules, or maybe not. Regardless, you should be able to correctly apply the rules with a little practice and careful thinking. The key to getting good at proofs is doing them, and learning actively from each so that you can build up your intuitions for what to do in different situations.

Let's put all of the pieces together by trying to prove the validity of the following argument.

1. $A \implies B$
2. $B \implies (C \vee \neg D)$
3. D
4. $A \wedge E$

$\therefore$ 5. C

This proof is going to be longer and a bit more complicated than some of the ones we will be doing in the early problem sets. But since we're doing this one together, I thought we could take a look at one with a few more steps. All of the tips and tricks will apply to simpler proofs.

STEP 1: Get your reference sheet and have it easily available. Nothing will slow you down more than not being able to double check the rules at a glance.

STEP 2: Always start every proof by re-writing and labeling the premises. This helps us to pay attention to what each says, and reminds us that we always have access to them.

1. $A \implies B$ [**Pr**]
2. $B \implies (C \vee \neg D)$ [**Pr**]
3. D [**Pr**]
4. $A \wedge E$ [**Pr**]

STEP 3: Read over the conclusion. You might even want to write the conclusion down at the top of the proof to keep it in your mind as the end goal. I usually just refer back to the question, but different people have different preferences.

STEP 4: Try to formulate a game plan for the proof. In this case, we could notice off the bat that the only occurrence of 'C' in the premises is in the consequent of line 2. This immediately gets me thinking that we will need to get access to this consequent by itself. This puts me on the look out for places where we could apply **MP**.

Seeing those early connections takes some familiarity, so whenever you feel stumped, go a different way. **Instead of trying to formulate a game plan this way, just start applying rules to simple lines where you see that it's possible.** We'll use this tactic this time around. In this particular proof, line 4 jumps out as a simple conjunction. We can use **Simp** on this line to generate two new lines. So far we have no idea if this will be helpful or not, but it's valid and so there's nothing stopping us from doing it.

5. A [4, **Simp**]
6. E [4, **Simp**]

That gives us these two lines. Remember, we need to cite our lines and rules whenever we introduce a new line into a proof. We now have two more lines at our disposal.

STEP 5: Any time we introduce a new line into a proof, quickly check the previous lines to see if we can use our new information. In this case, line 1 should jump out to us immediately. Line 5 is the antecedent of the conditional in line 1. Great, that means we get to use **MP**!

7. B [1,5 **MP**]

Now we repeat the process. Line 7 now allows us to use **MP** again.[31]

8. $C \vee \neg D$ [2,7 **MP**]

OK! Now we're cooking. We managed to generate a line that has our conclusion as part of it. Notice that we weren't being particularly deliberate in our approach. We just did what we could do with the available lines. We're close to the finish line now, but we have some work to do to extract the conclusion by itself from this disjunction.

Luckily, this particular example is one we already saw in our earlier discussion. First, we can take line 3 and apply **DN2**.

9. $\neg\neg D$ [3, **DN2**]

Almost done! One more line to go. Line 8 and line 9 can be used together with **DS**.

10. C [8,9 **DS**]

[31] Did you notice that it was also possible to use **HS** instead and **MP** once instead of using **MP** twice? There are usually multiple ways to do the same proof!

Line 10 corresponds with our conclusion. Since every line is either a premise, or derived from the premises using valid applications of the rules, this is a good proof and we are done! Congratulations, you've now done your first official natural deduction proof.

Before you head off to start working on problems by yourself, take note of some features of this proof. For example, we applied **Simp** to line 4, but then we never ended up using line 6 in the subsequent proof. So this proof has an extra line that we didn't need. It's no problem that it's there, but it takes up valuable time and space without helping the proof. As you get more comfortable, you may be able to tell when you don't need to apply all of the available rules.

Another thing to realize is that the order of the lines doesn't matter. As long as the last line of the proof is our conclusion, we can get there by any route. This flexibility is nice, but too many options can sometimes feel overwhelming. Taking things slow and following the steps listed above will go a long way towards taking the edge off of solving proofs.

2.6 Indirect and Conditional Proof Methods

Up to this point we've been exposed to a variety of different deduction rules. This has allowed us to show a wide variety of arguments and argument forms to be valid. We did this by taking a set of premises and directly applying our deduction rules to them until we arrived at the desired conclusion. If we followed all the rules, this would constitute a proof. Unfortunately, there is a limit to where these rules can take us. Take a look at the following very simple argument.

1. $\neg q$

$\therefore$ 2. $\neg(p \wedge q)$

First off, let's convince ourselves that this argument is valid. We could do this by drawing the truth table, but it should be pretty intuitive as well. If $\neg q$ is true, then how could a conjunction including q be true? q would be false, and so that conjunction would *always* be false. Hence, the argument is valid.

This is where we run into trouble from the perspective of a proof. Go ahead and glance at your rulesheet. Try to figure out how we could possibly prove that this is a valid argument (spoilers: at the moment, we can't!). Unfortunately, our rules are not comprehensive enough to allow us to construct a proof, which is a bit frustrating considering how simple the argument appears.

Luckily, we haven't exhausted all of the possibilities. While it's true that we can't solve this proof *directly*, we can try another, more *indirect* method. Indirect proofs work by introducing the ability to make **assumptions** into our proofs. We can then use these assumptions to indirectly prove our conclusions. In this section of the unit we'll look at two different methods, **Proof by Contradiction** and **Conditional Proof**.

2.6.1 Proof by Contradiction

Our first method has a Latin name, *reductio ad absurdum*, but our longstanding commitment to not learning Latin means that we'll just call it **Proof by Contradiction**.

The first thing we do in this indirect method is to assume that the conclusion of our argument is *false*. Of course, since this is a valid argument, we know that this is impossible. It is precisely this impossibility that we will be relying upon. Don't worry if this isn't making perfect sense as yet, it's much easier to understand when it's all said and done. For this argument, our proof currently looks like this:

1. $\neg q$ [**Pr**]
2. $(p \wedge q)$ [**ASSUMPTION**]

So far, all we have done is assume the negation[32] of what we are trying to prove. But now this gives us something else that we can work with in the proof. Let's see where the inclusion of this assumption takes us. We can now apply **Simp** to give us the following:

3 q [3, **Simp**]

Uh-oh! Have you spotted the issue as yet? Line 1 says $\neg q$, and now line 3 says q. If we put those two together, we would get $q \wedge \neg q$, which is a **contradiction**. However, as we know from **Unit 1**, a contradiction is *never* true! So what went wrong?

Turns out our meddling in the proof caused the contradiction. We only got this contradiction because *we* added the assumption in Line 2. That means that the assumption must be the cause of the later contradiction. And since we know that a contradiction is always false, our assumption must be false as well. This is the final piece of the puzzle. If the assumption is false, then its negation must be true. Thus, we have actually proven that the negation of the assumption is true, and we can enter it into our proof. This is why we specifically chose the negation of what we were trying to prove as our assumption, because now we have shown that the assumption is false, so its negation – our conclusion – is true.

[32] technically the negation of this conclusion is $\neg\neg(p \wedge q)$, but there is a good reason to make this assumption instead. I'll explain that in a little while.

This is an example of a proof by contradiction. The reason that this is an *indirect* method is that we haven't directly proved that the conclusion is true. Instead, we proved that its negation leads to a contradiction, and so its negation must be false. Our logical axioms took care of the rest, ensuring that the desired conclusion must be true.

When constructing proofs using this method, we use some special notation to make things easier to follow. In the future, when writing out the proofs, they will look like this:

1	$\neg q$		Pr
2		$p \wedge q$	As
3		q	Simp, 2
4		$\perp$	1, 3
5	$\neg(p \wedge q)$		

You should be able to read this proof, since it's the exact same as the one we just worked through. The difference here is that we use this style of writing out the proofs to keep track of where we made an assumption. We reserve the 'outermost' layer for the premises that we are given. In this example, that's $\neg q$ in line 1.

Next, we have our assumption. In order to make it obvious that this is where we are using the proof by contradiction, and keep it clear that the assumption is only 'true' in this segment of the proof, we create what's called a *sub-proof.* This is represented by the 'inner' layer. We mark the assumption with the justification 'As' to further reinforce that it's an assumption.

On line 4, we have this curious symbol '$\perp$'. In logic, this symbol represents a contradiction. The numbers on the right show which lines of the proof we get the contradiction from. This is what tells us that our indirect proof was successful, and allows us to 'close' the sub-proof and return to the outer layer, where we enter what we showed using the indirect method as a line on the proof. These are the usual steps when employing this sort of method.

2.6.2 Another example using Proof by Contradiction

Let's do another example to try to make the process a bit clearer. We will take things one step at a time. As we go, I'll relay some helpful tips for doing these kinds of proofs. First, the argument.

1. $D \implies C$
2. $D \vee A$

3. $\neg(A \vee B)$

$\therefore$ 4. C

Can you do this proof directly? You're free to try, but as we will see it's much easier when done indirectly. Sometimes, both a direct and indirect proof are possible, and it's just a matter of taking the path of least resistance. Next we will start the proof as usual by listing all of the premises.

1	$D \implies C$	Pr
2	$D \vee A$	Pr
3	$\neg(A \vee B)$	Pr

The next step is to make our assumption and create our sub-proof. In this case, the conclusion is C, so we will make the assumption $\neg C$. **N.B. If the conclusion of an argument has the form 'p', make the assumption '$\neg p$'. If it has the form '$\neg p$', make the assumption 'p'.**[33]

4	$\neg C$	As

We've now added our assumption. Remember that we do this by creating a sub-proof, aka a new layer to the proof. This reminds us that the assumption is only 'true' in this layer of the proof. From this point, we can just continue the proof like normal. Since we have $\neg C$, we can now apply **MT** using lines 1 and 4.

5	$\neg D$	MT, 1, 4

Can you see how the proof is shaping up? Remember that we are trying to force a contradiction to emerge. The simplest kind of explicit contradiction is always of the form '$p \wedge \neg p$'. Since we have a negated sentence on line 3, that seems like the easiest point of attack. Continuing on, let's apply **DS** to lines 2 and 5.

6	A	DS, 2, 5

There's only one more step now, because we can use **Add** on line 6 to create the contradiction. Here's the rest of the sub-proof.

7	$A \vee B$	Add, 6
8	$\perp$	3, 7

[33] In our first example we were trying to prove '$\neg(p \wedge q)$', so we used the assumption '$p \wedge q$', for example.

Line 8 tells us that there is a contradiction. The justification cites the relevant lines. Now that we have found a contradiction, we can close off the sub-proof and conclude that our assumption was false. Thus, the negation of our assumption is true.

9	$\neg\neg C$	IP, 4–8

Two important things to note here. First, we use the justification 'IP' for 'indirect proof', and we cite all of the lines of the sub-proof. Second, we need to be precise about what the sub-proof shows. It only allows us to conclude that the assumption is false, and so its negation is true. In this case, this leaves us with a doubly-negated sentence. Even though it's trivial to now apply **DN** and achieve our desired conclusion, natural deduction doesn't allow for any shortcuts. Therefore, we can finish off the proof like so:

10	C	DN, 9

And there we go. Here is what the full proof would look like if we did it all at once. Try to spend some time familiarizing yourself with this proof. Make sure you understand why every step of the proof is important, and how you would go about doing it yourself.

1	$D \implies C$	Pr
2	$D \vee A$	Pr
3	$\neg(A \vee B)$	Pr
4	$\quad \neg C$	As
5	$\quad \neg D$	MT, 1, 4
6	$\quad A$	DS, 2, 5
7	$\quad A \vee B$	Add, 6
8	$\quad \bot$	3, 7
9	$\neg\neg C$	IP, 4–8
10	C	DN, 9

2.6.3 Conditional Proof

Now that we have been introduced to indirect proofs by contradiction, conditional proofs will come more naturally to us. This is a method that we can use whenever we want to introduce a conditional into our proofs. We have actually already used this method once, but informally. We did this when we were trying to intuitively justify our rule **HS**. Just to refresh your memory, here is what that rule said:

$$\begin{array}{l} 1.\ p \implies q \\ 2.\ q \implies r \\ \hline 3.\ p \implies r \end{array}$$

When we discussed this rule, I gave some informal reasoning about its inclusion. However, it is possible to prove that this rule is valid. In order to do that, though, we need to introduce the concept of a **conditional proof**.

A conditional proof is a bit like an indirect proof, in that we make an assumption and use it to show something in a sub-proof. The differences are with respect to what we choose as our assumption, and what we are trying to show. As the name suggests, a conditional proof tries to prove a conditional.

We start off by assuming or hypothesizing that the antecedent of the conditional is true. This is a pretty natural thing to do, after all, because a conditional is an 'if...then' statement, so all we're doing is pretending that the antecedent is true. Once we do that, if we are able to show something else is true, we will have proven that *if* the antecedent is true, *then* the consequent is true as well, aka we will have shown that a conditional is true. Let's see it in action.

1	$p \implies q$	Pr
2	$q \implies r$	Pr

So far we've just restated the premises. Now is where the conditional proof comes in. We are trying to prove '$p \implies r$', so for our assumption we should choose the antecedent, i.e. 'p'. From here, the proof is really simple, just two successive applications of **MP**.

3	p	As
4	q	MP, 1, 3
5	r	MP, 2, 4

What this tells us is that *if* our assumption p is true, *then* we can show that r must be true as well. But that's all the conditional $p \implies r$ means! As a result, we can now close off our sub-proof and conclude that the conditional is true, just as we want.

6	$p \implies r$	CP, 3, 5

Conditional proofs are a powerful way to introduce conditionals into a proof. In the next example, we'll go through another proof that uses conditional proofs, only this time we will show how introducing the conditional in the *middle* of the proof can be worth our while.

2.6.4 Another Example Using Conditional Proof

Let's use the method of conditional proof to prove a biconditional. As we will see in the final section of the unit, this is a very common application of conditional proofs. Proving a biconditional is nothing more than proving two different conditionals, so there's no new concepts. The only unfamiliar thing to keep track of is the sub-proofs. Make sure you're keeping an eye out for when one sub-proof ends, and another begins. Here is the argument:

1. $A \implies \neg C$
2. $\neg B \implies C$
3. $A \vee \neg B$

$\therefore$ 4. $A \iff B$

Because the conclusion is a biconditional, we can prove it by proving $A \implies B$ and $B \implies A$ and then using rule **Equiv**. We start the proof as normal, listing our premises.

1	$A \implies \neg C$	Pr
2	$\neg B \implies C$	Pr
3	$A \vee \neg B$	Pr

Next, we want to introduce our first assumption. Let's prove the conditional $A \implies B$ first.[34]

[34] This time I won't go through every single step. You should be able to follow the sub-proof easily enough, since it's like the proofs in Problem Set 4.

4	$\quad A$	As
5	$\quad \neg C$	MP, 1, 4
6	$\quad \neg\neg B$	MT, 2, 5
7	$\quad B$	DN, 6
8	$A \implies B$	CP, 4–7

This is the end of the first sub-proof. Now we close off that sub-proof and end the conditional proof, giving us our first conditional. But now we need to prove $B \implies A$. That means that we need to start a new sub-proof with a different assumption. This is what it would look like.

9	$\quad B$	As
10	$\quad \neg\neg B$	DN, 9
11	$\quad A$	DS, 3, 10
12	$B \implies A$	CP, 9–11

At this point we're basically done, it's just a matter of adding the final line to the proof.

13	$A \iff B$	Equiv, 8, 12

That's it! We have successfully used two applications of **CP** in the same proof to prove a biconditional. Hopefully you are starting to see just how powerful these techniques are. In the final section of the unit we'll put these methods to great use. But before that, pause here and spend some time practicing these two new methods on **Problem Set 5**. For reference, here are the two rules listed out like the others. You'll also find this on the updated rule sheet.

Indirect Proof

1	$\quad p$
$\vdots$	$\quad \vdots$
n	$\quad \bot$
$n+1$	$\neg p$

Conditional Proof

1		p
$\vdots$		$\vdots$
n		q
$n+1$	$p \implies q$	

2.7 Putting Everything Together: Derived Rules and Complex Proofs

We have learned many different natural deduction techniques and applications. With the introduction of conditional proofs and proofs by contradictions, we are now in a position to prove some powerful results. In particular, we can use these techniques to prove some additional rules – what we call *derived* rules – that will then make our lives easier in the future.

Usually, these derived rules take the form of logical equivalences between two different sentence forms. If two sentence forms are logically equivalent, then it is possible to prove this logical equivalence starting from zero premises. That may sound surprising to you now, since we've relied on premises in the past, but clever uses of our new techniques will make this process much easier. Let's begin with a logical equivalence we saw way back in our initial discussion of conditionals.

We will now prove that $(p \implies q) \iff (\neg q \implies \neg p)$ starting from zero premises. We will do this just like the earlier example of proving a biconditional, by using two applications of **CP** to prove two conditionals, and then using **Equiv**.

1	$p \implies q$	As

This starts off the proof. However, it seems like we don't have anywhere to go. After all, we have no extra premises to work with. But if you take a close look at what we're trying to prove, you'll notice that it's *another* conditional. That means that we can use another conditional proof, nested inside the first one, to prove this second conditional. Again, this emphasizes the importance of sub-proofs and keeping track.

1	$\quad\quad p \implies q$	As
2	$\quad\quad\quad\quad \neg q$	As
3	$\quad\quad\quad\quad \neg p$	MT, 1, 2
4	$\quad\quad \neg q \implies \neg p$	CP, 2–3
5	$(p \implies q) \implies (\neg q \implies \neg p)$	CP, 1–5

This is the first half of the proof. Do you see how we used **CP** twice? We're going to do the same thing again, this time for the reverse conditional.

6	$\quad\quad \neg q \implies \neg p$	As
7	$\quad\quad\quad\quad p$	As
8	$\quad\quad\quad\quad \neg\neg p$	DN, 7
9	$\quad\quad\quad\quad \neg\neg q$	MT, 6, 8
10	$\quad\quad\quad\quad q$	DN, 9
11	$\quad\quad p \implies q$	CP, 7–10
12	$(\neg q \implies \neg p) \implies (p \implies q)$	CP, 6–11
13	$(p \implies q) \iff (\neg q \implies \neg p)$	Equiv, 5, 12

This completes the proof. Starting from no premises whatsoever, we were able to prove this biconditional. The significance of starting from zero premises is that we could insert this proof into any other proof if we wanted. That's very useful for us, because it means that we can now use this as the basis to 'swap' between those two conditional forms if one would be easier to work with than the other. Instead of having to repeat the whole process of proving this biconditional every time, we just come up with a shorthand name – a new rule of deduction.

Rule Contra:

1. $p \implies q$

---

2. $\neg q \implies \neg p$

Rule Contra:

1. $\neg q \implies \neg p$

---

2. $p \implies q$

Let's prove one more derived rule. This is one of a pair of rules called **DeMorgan's Laws**. You will prove the other one in **Problem Set 6**. We will now prove that $\neg(p \wedge q) \iff (\neg p \vee \neg q)$. In this example, I'll do the entire proof all at once. Use it as a test to make sure you can follow along with the steps of a complex proof and keep track of all of the sub-proofs.

This proof gives us the following two rules which we can use in later proofs. As you can hopefully see, we are using natural deduction to great effect, building up our arsenal of rules to make more complex derivations easy by comparison.

Rule DeM:

1. $\neg(p \wedge q)$

---

2. $\neg p \vee \neg q$

Rule DeM:

1. $\neg p \vee \neg q$

---

2. $\neg(p \wedge q)$

1	$\neg(p \land q)$	As
2	$\neg(\neg p \lor \neg q)$	As
3	$\neg p$	As
4	$\neg p \lor \neg q$	Add, 3
5	$\bot$	2,4
6	$\neg\neg p$	IP, 3–5
7	p	DN, 6
8	$\neg q$	As
9	$\neg p \lor \neg q$	Add, 8
10	$\bot$	2,9
11	$\neg\neg q$	IP, 8–10
12	q	DN, 11
13	$p \land q$	Conj, 7, 12
14	$\bot$	1,13
15	$\neg\neg(\neg p \lor \neg q)$	IP, 2–14
16	$\neg p \lor \neg q$	DN, 15
17	$\neg(p \land q) \implies (\neg p \lor \neg q)$	CP, 1–17
18	$\neg p \lor \neg q$	As
19	$p \land q$	As
20	p	Simp, 19
21	$\neg\neg p$	DN, 20
22	$\neg q$	DS, 18, 21
23	q	Simp, 19
24	$\bot$	22,23
25	$\neg(p \land q)$	IP, 19–24
26	$(\neg p \lor \neg q) \implies \neg(p \land q)$	CP, 18–25
27	$\neg(p \land q) \iff (\neg p \lor \neg q)$	Equiv, 17, 26

This marks the end of the unit's notes, but not the end of the unit. Proofs are all about practice and familiarity with the rules. Go over these proofs as many times as you need for the steps to make sense, then go out and give **Problem Set 6** a try yourself. Don't get discouraged if it's taking a while. Any intermediate logic student will tell you that there's a mountain of scrap paper full of wrong attempts hiding behind their pristine final proofs. There is a certain amount of trial and error – seeing which assumptions work and which are dead ends. Persistence pays off, and the sense of fulfillment from getting out a difficult proof is always worth it!

3 Unit 3: Introduction to Predicate Logic

3.1 Fixing the Gaps in our Translations

In Unit 1, we learned how to translate English sentences into sentential logic notation. After an entire unit on proofs, hopefully you will agree that this was a very helpful tool, making it much easier for us to prove that many long and complicated arguments were valid, in addition to proving many intricate logical equivalences without having to resort to truth tables. However, with that being said, sentential logic left some gaps in what we are able to accomplish relative to the complexities of English.

For example, let's consider a sentence we translated sententially in Unit 1: "Dustin is quite lazy, while Eileen and Faried are hard workers." In Unit 1, we translated this sentence using the following kind of translation key:

D = "Dustin is a hard worker."[35]
E = "Eileen is a hard worker."
F = "Faried is a hard worker."

After that, we would have translated this whole sentence as "$\neg D \wedge (E \wedge F)$." Don't you find this kind of translation unsatisfying? I think it is, because it misses out on the logical connections between the atomic sentences. It fails to demonstrate that there's something about Dustin – some feature of his – that is the opposite of Eileen and Faried (who are similar in this regard). Back in Unit 1, I said that logic isn't so concerned with meanings as with truth. That's still true, but what if there was a way to show the logical connection between atomic statements without sacrificing anything?

Similarly, consider the first example that I ever gave in this course of a valid argument:

[35] You might not have done this, since it's not obviously the case that being lazy is the negation of being a hard worker. In fact, it might not be strictly true. But ignore all of that, I'm trying to make a dramatic illustration!

1. All humans are mortal.
2. Socrates is a human.

∴ 3. Socrates is mortal.

Now let's translate this argument using what we have learned. All three of these sentences are atomic statements, so let's just pick a different letter for each. It looks like:

1. A
2. B

∴ 3. C

Have you noticed the problem? This argument is obviously valid – there's no way for the premises to be true while the conclusion is false. However, our sentential translations don't show that this is the case. In fact, they *couldn't*. There's simply no way for sentential logic to show the validity of a simple argument like this. That's a big problem!

The last kind of example concerns a use of the word 'and' that was mentioned in Unit 1. Consider a sentence like "Jane and Karen are friends." As I mentioned in Unit 1, it's intuitively wrong to describe this sentence as saying that "Jane is a friend and Karen is a friend", because it clearly means that Jane and Karen are friends *with one another*. In Unit 1, all we could say is that we would translate a sentence like this as an atomic sentence. But is that really all we can do, from a logical perspective, to describe the structure of a sentence like this? The answer – for all of these examples – is 'no'. There is in fact a system of translation that allows us to show all of the features of the above examples that sentential logic misses out on. That system is called **Predicate Logic**, and is the topic of this last unit of the class. We will see how to translate these sentences (and many others like them).

3.2 Introduction to Predicates and Individuals

In this section we will begin by introducing the new smallest unit of language that we can translate. In sentential logic, that's a sentence. In predicate logic, we have what's called a logical *predicate*:

Definition: A **predicate** is a property or feature of an object or objects. It describes an attribute that can be associated with individuals. A predicate is always represented by an uppercase letter.

What counts as a predicate? Really, it's any property or feature that you can ascribe to something. For example, "the property of being smart"; "the property of being born in Oklahoma"; "the property of owning exactly twelve pairs of shoes." These are just some random examples of what can count as a predicate. These are the new smallest units of our translations. However, just describing properties does not get us very far, because we need to be able to describe to *who or what* the predicates apply. In order to do this, we take every object under consideration[36] and we assign symbols to those objects. We call these *individual constants.*

> **Definition:** An **indivdual constant** is a symbolic name that is given to a particular individual in the domain of discourse. Every individual has a unique constant, and constants are always written as lowercase letters.

As usual, an example will help us to see what's really going on here. Let's go back to the sentence I mentioned in the introduction, "Dustin is quite lazy, while Eileen and Faried are hard workers.". Before, the most specific way that we could translate this sentence was as "$\neg D \wedge (E \wedge F)$." Now that we have the notion of predicates and individual constants, we can do much better. We will use the following translation key:

$Hx = x$ is a hard worker.[37]
$d =$ Dustin
$e =$ Eileen
$f =$ Faried

This gives us a translation key for the sentence, which we can now translate as: "$\neg Hd \wedge (He \wedge Hf)$". As you can tell, in order to show that a particular individual has a certain property, we put the constant name for the individual immediately after the predicate symbol. Therefore, we read something like "$\neg Hd$" as "It is not the case that Dustin is a hard worker".

One thing to pay attention to with these translations is how predicates can come one right after the other in English. For example, the sentence "Christian is a short person" says both that Christian is short and that he is a person. Therefore, if we were going to translate this sentence into predicate logic, we would need to use two predicates, one for 'short' and one for 'person': '$Sc \wedge Pc$'.

Before, our translations left out many of the logical connections inherent in each of our sentences. Now that we are armed with the concepts of predicates and names for individuals, we are in a much stronger position with respect to our

[36]In just a short while we will discuss which objects count as being under consideration in a specific context.
[37]Ignore the variable name 'x' for now, we will explain how these work in the next section.

translations.

3.2.1 Practicing Simple Predicate Logic Translations

This section is just going to be a few short examples to get you acquainted with looking at basic predicate logic translations. Make sure that you understand why every sentence gets translated the way that it does to shore up your basic knowledge.

1. "Amanda and Brandon enjoy skiing."
   $Sa \wedge Sb$

2. "Either Charmaine is tall or she owns a ladder."
   $Tc \vee Lc$

3. "If Derek eats dairy, he gets a stomachache."
   $Dd \implies Sd$

4. "Edith is excellent at English and Frida is great at chemistry, but neither of them are proficient at mathematics."
   $(Ee \wedge Cf) \wedge (\neg Me \wedge \neg Mf)$

5. "Gary is enjoying himself if and only if he's at the beach."
   $Eg \iff Bg$

3.3 Quantified Statements

In the previous section we saw how basic predicate logic worked. Now we can translate sentences about individuals and their features in a much more sophisticated way. But what about simple arguments like the one mentioned in the introduction?

1. All humans are mortal.
2. Socrates is a human.

$\therefore$ 3. Socrates is mortal.

Something about this argument is still eluding our grasp. Sure, we could translate 2. as 'Hs' and 3. as 'Ms', but we don't know how we might translate a sentence like 1., even using basic predicate logic. In order to be able to translate a sentence like this, we need to introduce the notion of a *quantifier*, which gives the ability to talk about objects and individuals *in general*.

There are two kinds of quantifiers that we will be introducing in this section, the **universal** quantifier and the **existential** quantifier.

3.3.1 The Universal Quantifier

The first quantifier that we will look at is called the universal quantifier. It gives us a way to talk about all of the objects in our domain at the same time. But before we discuss how it works, we should finally clarify what it means to talk about a *domain* in logic.

Definition: A **domain** is a defined set of objects. When a domain isn't specified, the domain defaults to all of the objects in the universe, or *universal set*.

As usual this definition will only make perfect sense after applying it to an example. Here's a fun one. Suppose that you feel for a bowl of cereal and you notice your housemate is in the kitchen, so you ask them "Is there any milk left?" Then, in reply they say to you "Yeah, probably, but there's none in the fridge!" If this happened to you, you'd be understandably annoyed with your housemate, because it's obvious from the context that what you asked was if there was any milk left *in the fridge*.

In this case, you were implicitly (i.e. without actually saying it) restricting the domain of the conversation to just the fridge. What made your housemate's response an annoying wisecrack was that they were using a different domain (everything) than the one you intended. That's the basic idea of a domain – it specifies what set of objects and individuals are being discussed in any particular instance.

For our purposes, the domain is unrestricted by default, i.e. it covers every object in the universe. However, it is possible to restrict the domain of discourse by stating what the new domain is for that particular context.

Now that we understand the concept of a domain, we are in a position to define the universal quantifier. What the universal quantifier does is it allows us to make statements about all of the objects in the domain. In order to do this, for example we say "For all/for every object x, Px" where x is an *object variable*, and 'P' is a predicate. Just like with sentence forms in Unit 1, we use object variables in predicate logic to talk about unknown objects, or objects in general.

We also have special notation for the universal quantifier, so that we are not stuck writing "For all/for every" every time we want to write a universal sentence. Instead, we use the symbol '$\forall$', which means the same thing as 'For all/for every'.

Now how about sentence 1. above? Let's use the universal quantifier to translate that sentence into predicate logic. First, we assign predicate letters to each of the predicates. As you might expect, we'll use 'Hx' for 'x is human' and 'Mx' for 'x is mortal'.

How do we translate a sentence like "All humans are mortal"? Well, first of all, we know that the sentence is a universal one, so we begin with the universal quantifier, saying "$\forall x$". What do we want to say about these objects? We are trying to say that any object in the domain that is a human is also an object that is mortal. Pause for a second and see if you can guess how you would translate a sentence like this into predicate logic.

It might not be intuitive, but sentences of the form "All A are B" are translated as *conditionals*. In this case, we get $\forall x(Hx \implies Mx)$, or "For all x, if x is a human then x is mortal."

Don't worry if this translation is a bit puzzling right now. Once we done covering both kinds of quantifiers, we will return to these kinds of sentences in much more depth. For now, though, it should be much easier to tell that our original argument is valid using the predicate logic translation.

1. $\forall x(Hx \implies Mx)$
2. Hs

$\therefore$ 3. Ms

In the next sub-section we will take a look at the other quantifier, the existential quantifier.

3.3.2 The Existential Quantifier

The system of predicate logic isn't perfect. Translating vague words like 'most' and phrases like 'a few' are basically impossible without some kind of arbitrary stipulation. However, our system of predicate logic is at least good enough to translate some kinds of quantities. We have already seen how to do this for universal 'all' or 'every' statements. Next, let's see how to translate sentences about 'some', or what's called an 'existential' claim.

Consider a sentence like "Someone is happy with their grades." What does a sentence like this say exactly? You might think that the word 'some' means something like 'at least a few', but in logic this isn't the case. Think about it like this: if at least one student is happy with their grades, isn't it true that *some* of the students are happy? One student is still some of them, even though it's a bit misleading to use the word 'some' when we know that it means 'just one'. Nevertheless, we are only interested in what's true and false and not what might be misleading.

N.B Therefore we always translate the word 'some' to mean 'at least one'.

That's where the existential quantifier comes in. We use the symbol $\exists$ to mean 'there exists' or 'there is'. Thus, we can translate a sentence like "Someone is happy with their grades" as "$\exists x(Hx)$" which says "There is an x such that x is happy with their grades", which says the same thing as "Someone is happy with their grades."

With the introduction of the two new quantifiers, we are now ready to jump into translating a variety of different sentences into predicate logic. Some translations are trickier than others, so it will be helpful to go through a few examples of each.

3.4 Updated Rules for Well-formed Formulas

Before we get started on translations, it is important for us to re-visit what counts as a wff now that we have the two additional quantifiers. This is vital for both predicate logic translations and proofs, since changing the scope of a quantifier dramatically alters its meaning.

Definition: A **well-formed formula ('wff')** of predicate logic is any formula that consists of:

1. An atomic statement, given by the combination of a predicate, and either an individual name, 'n', or a variable 'x, y, z' as in 'Pn, 'Qy;
2. If A is a wff, then $\neg A$ is a wff;
3. If A and B are wffs, then $(A \wedge B)$ is a wff;
4. If A and B are wffs, then $(A \vee B)$ is a wff;
5. If A and B are wffs, then $(A \implies B)$ is a wff;
6. If A and B are wffs, then $(A \iff B)$ is a wff;
7. If A is a wff, then $\forall x A$ is a wff;
8. If A is a wff, then $\exists x A$ is a wff;
9. Any construction that cannot be obtained just by repeated applications of rules 1-6 is **not** a wff.

The preceding box tells us that wffs in predicate logic are basically the same as for sentential logic, except that what counts as an atomic statement has changed, and two new kinds of sentences count as well-formed formulas. All of the other rules are the same. As we keep going forward, you'll gain a lot of practice translating these kinds of sentences, and applying the correct scope to the quantifiers will feel more second-nature.

3.5 Translating Universal Statements

In the earlier section where we introduced the universal quantifier, we translated the sentence "All humans are mortal" as "$\forall x(Hx \implies Mx)$". Perhaps this translation does not seem intuitive to you. In this section, we'll go through a number of examples and the rules of thumb associated with them so that the translations come more naturally.

Let's cover statements with the same general form as the one above, "All Ps are Qs". **The rule of thumb for these kinds of statements are that they all get translated as conditional of the form $\forall x(Px \implies Qx)$.**

You might be asking yourself why these translations come out as conditionals. I think one of the easier ways to see why this is the case is to think in terms of *sets* or *groups*. Think of what a sentence like "All cats are mammals" communicates to us. It is saying that the group or set of all the cats is contained within the set of all the things that are mammals. There are other mammals besides cats, but there aren't any cats that aren't mammals.

So "All cats are mammals" is saying something like "Any object that is in the set of cats must also be in the set of all mammals". In other words, any time it's *true* that something is a cat, it must also be *true* that the thing is a mammal. On the other hand, if something is a cat, but it isn't a mammal, then "All cats are mammals" has got to be false. These are the same truth conditions as for the conditional, which is one way to remember that "All Ps are Qs" is translated as a conditional every time.

Furthermore, all of these sentences have the same translation as "All Ps are Qs":

1. "Every P is a Q."
2. "Anything that is P is also Q".
3. "It is true for everything that if something is P, then it is also Q".
4. "Whenever something is P it is Q."

It might seem a bit counter intuitive, but these rules of thumb are very consistent, so you will do well to internalize them and repeat them to yourselves.

One of the biggest pitfalls to avoid when translating universal statements is in misinterpreting the scope. For example, consider the following two sentences:

"All humans needs to eat"; and "If everything is human, then everything needs to eat".

As we just saw, the first sentence would get translated as '$\forall x(Hx \implies Ex)$'. However, this is not true for the second sentence. Notice that the *scope* of the universal quantifier in the first sentence covers the entirety of the remainder of that sentence, but in the second sentence there are *two* different universal quantifiers, corresponding to the two different uses of 'everything'. That means that the second sentence would be translated as "$\forall x(Hx) \implies \forall x(Ex)$".

N.B. Quantifiers, like negations, always have scope over the smallest complete statement that comes immediately after them.

These two sentences say different things and **are not interchangeable**. The first sentence says that all of the things in the universe that are humans are also things in the universe that needs to eat. The second sentence says that *if* everything in the universe is human, *then* everything in the universe needs to eat. It is very important that you keep track of the locations of the quantifiers in the sentences so that you get an idea of their scope. If you give a quantifier the wrong scope you will say something that is either stronger or weaker than what was intended.

Here are a few examples of different universal statements. Try to translate them yourself and then check the footnote for the answers (don't worry if you've used different predicate letters or variables, just focus on the scopes of the quantifiers).

1. "All of the students in the class are happy."
2. "If everyone is partying, then everyone is happy."
3. "Whenever all the students are happy, they get good grades."[38]

There will be more practice questions at the end of the whole section on translations, so don't get too stressed about it now. The next thing is to move on to translating different kinds of existential claims.

3.6 Translating Existential Statements

Existential statements are translated differently than universal statements, so learning the general rules for both is important. If the 'standard' universal statement is "All Ps are Qs", then the standard existential statement is "Some

[38]
1. $\forall x(Sx \implies Hx)$
2. $\forall y(Py) \implies \forall y(Hy)$
3. $\forall x((Sx \wedge Hx) \implies Gx)$

Ps are Qs". However, as we discussed earlier, saying 'some' in a logical context is identical to saying 'at least one', so we could also write this sentence as "At least one P is Q".

Earlier, we saw that basic universal statements are translated as conditionals. However, this is not the case for existential statements. When we say "At least P is Q", what we are trying to say is that something exists that has *both* of the properties P *and* Q. The italicized words are the operative ones in this case.

N.B. Basic existential claims of the form "Some Ps are Qs" are translated as $\exists x(Px \wedge Qx)$.

Because basic existential claims are normally translated as conjunctions, they're a fair bit easier to remember than the translations for universal statements. Nonetheless, try to commit the rule to memory for the next section where we add negations into the mix.

3.7 Mixing Quantifiers and Negations

This is where things start to get quite tricky. There is one major problem when it comes to translating quantified sentences with negations. The problem has to do with scope. Both negations and quantifiers have very different meanings when their scopes change. All the way back in Unit 1, we used the following two sentences to discuss the scope of negations.

a "Everyone in the class is not happy."

b "Not everyone in the class is happy."

In Unit 1 we only had the knowledge to translate these into sentential logic, so we utilized (a) and (b) to demonstrate that only (b) counted as a negation of the sentence "Everyone in the class is happy". But now that we have moved on to doing predicate logic, we can now correctly identify that these are two universal statements. That means that we should be able to use predicate logic to translate them, but in order to do that we need to pay attention to the scopes of the quantifiers.

In (a), notice that the word 'everyone' – which indicates the universal quantifier – comes before the word 'not' – which indicates negation. This order is reversed in (b). When translating these sentences we must preserve the order in which these logical operators appear.

In (a), the universal quantifier is all the way at the beginning of the sentence, meaning it should have the widest scope. Thus, we would expect the sentence to look like '$\forall x(A)$' for some wff A. If we look at the position of the negation, we see that it is immediately before the word 'happy'. That means it has a very narrow scope, covering just that predicate. Therefore, the translation should look like

(a) $\forall x(Cx \implies \neg Hx)$

On the other hand, the negation in (b) has a wider scope than the quantifier. Therefore, we would expect this sentence to look like '$\neg A$' for some wff A. The rest of the sentence has the basic form of a universal statement "All Ps are Qs", so we can just translate it in the usual way, and then place a negation on the outside, as in

(b) $\neg\forall x(Cx \implies Hx)$

Notice the way that the quantifiers and negations in each case maintain the same relative positions in the translations as they have in the original sentences. Use this scheme to keep track of your translations. Remember every time to check the positions of the negations and quantifiers.

There is a lot more to say about quantifiers and negations, however. Consider a sentence like "No one in the class is happy" and compare it to sentences (a) and (b). This sentence has the same translation as one of those, but which one is it? Take a moment to guess.

The answer is (a). This may come as a surprise, but it is true. What this means is that "No one in the class is happy" is *not* the negation of "Not everyone in the class is happy". The best way to see this to investigate the truth conditions for each of the sentences.

Suppose the class consists of three students, Albert, Betty, and Charlie. Now suppose that I tell you that Albert and Betty are happy, but Charlie is not. Is it true that "Not everyone in the class is happy"? It clearly is, because Charlie is unhappy, so not everyone is happy. But will it be true that "No one is happy"? Of course not! We are supposing that both Albert and Betty are happy. Thus, it should be clear that the negation of a universal statement is not a 'none' statement, but instead a 'not all' statement. A 'not all' statement says something that is logically weaker than a 'none' statement.

There's even more, though! In the previous example, we showed that 'Not everyone in the class is happy' would be true so long as *at least one* member of the class was unhappy. 'At least one' is one of our indicator words for an existential claim, though, so that means that the sentence "Not everyone in the class is happy" is logically equivalent to the existential sentence "At least one person is not happy".

$\neg\forall x(Cx \implies Hx) \iff \exists x(Cx \wedge \neg Hx)$

However, this expression of the existential claim obscures what happens when you swap between a universal and existential statement. Let's express the existential statement in a different form using some more logical equivalences.

First off, notice that '$Cx \wedge \neg Hx$' is logically equivalent to '$\neg(\neg Cx \vee Hx)$' by DeMorgan's law. Next, we can also notice that '$(\neg Cx \vee Hx)'$ is logically equivalent to '$Cx \implies Hx$' by Implication. Therefore, $\exists x(Cx \wedge \neg Hx)$ is logically equivalent to '$\exists x\neg(Cx \implies Hx)$'. So we get the following logical equivalence:

$\neg\forall x(Cx \implies Hx) \iff \exists x\neg(Cx \implies Hx)$

Take a close look at the relationship between these two statements. To go from one to the other, we move the negation to the other side of the quantifier, and then swap the quantifiers. This relationship doesn't just hold for these two sentences. It's a general logical equivalence that is kind of like DeMorgan's Laws, but for quantifiers. We'll call them 'quantifier negation' (QN) rules. There are two of them:

QN	**QN**
1. $\neg\forall x(A)$	1. $\exists x\neg(A)$
2. $\exists x\neg(A)$	2. $\neg\forall x(A)$
QN	**QN**
1. $\neg\exists x(A)$	1. $\forall x\neg(A)$
2. $\forall x\neg(A)$	2. $\neg\exists x(A)$

These equivalences are massively helpful for doing proofs in predicate logic as rules of inference. However, they are still very useful for remembering the relationships between negated existential and universal claims. Let's do another example to show how to switch between different quantifiers.

We'll start with the existential sentence "It is not the case that some people are lizards"[39]. We start off by translating this sentence using an existential quantifier. The phrase 'it is not the case' signals a negation, and since it comes at the beginning of the sentence, we give it the widest scope. Next, we have the basic existential 'some people are lizards', so we can translate this in the usual way. This yields

$\neg\exists x(Px \wedge Lx)$

Next, we want to transform this existential statement into a universal one. In order to do this, we move the negation to the other side of the quantifier and then swap the quantifier. This gives us

$\forall x\neg(Px \wedge Lx)$

The last thing to do is to try to verify in English that this sentence has the same logical meaning as the one that we started with. This sentence says "Everything is not both a person and a lizard". Hopefully it's clear to see that this sentence will be true if and only if "It is not the case that some people are lizards" is true.

Before we close off this section, here are some practice problems that you can work on. You'll find the answers to these questions towards the end of the unit. If you translate a sentence as a universal and the solution is written as an existential (or vice versa) try to use what we learned in this section to switch between quantifiers and then re-check your answer.

1. "All football players are athletes, but not all athletes are football players."
2. "Anyone who studies passes the course."
3. "Some cookies have raisins."
4. "No eager students procrastinate."
5. "If anyone fails the course, then some will be sad."

3.8 Relational Predicates

At this point we've resolved some of the lingering concerns that we had in Unit 1 concerning translations. However, there is one more case left to go. In the section on conjunctions, I mentioned that the following sentence couldn't be translated as a conjunction:

[39] This is true, despite what some zany conspiracy theorists will tell you about various world leaders.

(*) "Jane and Karen are friends."

The reason for that is that "Jane is a friend and Karen is a friend" is *not* a correct translation of what (*) is expressing. What (*) *does* say is that Jane and Karen are friends *with each other*. Fortunately, predicate logic gives us a way to translate a sentence like (*). In order to do this, we need to develop the concept of a **relation**.

Simply put, a relation is a predicate that compares or relates more than one object to one another. So far, all of the predicates that we have seen are 1-place predicates, which means that they only describe one object or individual. For instance, 'the property of being a person' or 'being taller than six feet'. A relation, on the other hand, is a 2-place predicate. It relates two objects to one another. For a bit of a visual representation, we can imagine a 1-place predicate looking like

P

What goes in the blank space is the name of an object in the domain. So if P = 'being proud' and a = Alfred, we could write 'Alfred is proud' as 'Pa'. Therefore, we can think of a relation as a 2-place predicate, or one that has *two* blanks instead of one.

Now let's go back to (*). In order to translate 'Jane and Karen are friends', we need to consider the friend relation. If we write 'Lxy', that would mean 'x is friends with y'. So if we want to translate (*), we could write

(*) = 'Ljk'

Relations like this are easy. Here are some examples for practice.

1. "Anna and Beatrice are sisters."
2. "Celia and Dante are in love."
3. "Eugene is Frank's parent."

3.9 Symmetric, Transitive, and Reflexive Relations

Relations are very interesting from a logical perspective, and can have many different features. In this section we will talk about some of those features, and use them to reinforce our concept of a relation.

3.9.1 Symmetric Relations

Some relations have the property of being **symmetric**.

Definition: A relation R is **symmetric** if $\forall x, \forall y\ Rxy \iff Ryx$.

Don't be alarmed by this kind of definition, it's really easy to explain. As an example, think of the relation 'being married to'. If it's true that Grace is married to Haley, then it's also true that Haley is married to Grace. In fact, it couldn't be false. So for the relationship of being married, $Rgh \iff Rhg$.

However, not all relations have this property. Consider the relation 'being a parent'. If Tom is the parent of Sally, then it's *not* true that Sally is the parent of Tom. In this case, $Pts \wedge \neg Pst$.

It's important to consider whether or not a relation is symmetric when doing translations, because if a relation is not symmetric and you write the order of the names wrong then you will have failed to correctly express the logical meaning of a statement.

3.9.2 Transitive Relations

Some relations have the property of being **transitive**.

Definition: A relation R is **transitive** if $\forall x, \forall y, \forall z$ if Rxy and Ryz, then Rxz.

A transitive relation is basically one that 'transfers' between individuals. As an example, consider the relation of 'being the descendant of'. If a is the descendant of b, and b is the descendant of c, then a is the descendant of c. In pure notation, $(Dab \wedge Dbc) \implies Dac$.

Now we need to be a bit careful about some relations. The 'descendant' relation, for instance, is transitive but not symmetric. However, some relations can be both transitive and symmetric at the same time. For instance, the relationship of being in the same room is both transitive and symmetric. Make sure to pay attention to the kinds of relations that are being used so that you don't make any mistakes with the order of the individuals.

3.9.3 Reflexive Relations

The last property of relations that we'll discuss is **reflexivity**.

Definition: A relation R is **reflexive** if $\forall x$, Rxx .

Reflexive relations are ones that hold between individuals and themselves. One easy example of a reflexive relation is 'being identical with'. Everything is identical with itself!

These are some small examples of the various ways that relations can work. Now all that is left is to put all of these ideas together to translate quantified sentences with negations.

3.10 Multiple Quantifiers and Relations

We have been slowly building up our ability to translate more and more complex statements as the unit has gone on. The last kind of sentence that's worth talking about is one that features both relations and quantifiers at the same time. We'll start off slow. Consider a sentence like 'Abigail loves someone'. Hopefully it's not too difficult to see that we would write such a sentence as

$\exists x(Lax)$, or in other words "There exists someone such that Abigail loves them".

But how about a more complex statement like "*Everyone* loves *someone*"? These translations are much more difficult. In the first place, notice that this sentence is technically ambiguous. Is it saying that everyone loves someone or other? Or is it saying that there is a particular person such that everyone loves that one person? I think most people would default to the former interpretation. Fortunately for us, predicate logic can translate both of these ideas in a way that makes it clear which meaning is intended. The key to doing this is by correctly ordering the scope of our quantifiers.

Let's say that we want to translate "everyone loves someone or other". We start by paying close attention to the order of the quantifiers in the sentence. Since 'everyone' comes first, it is likely that the universal quantifier has the widest scope. So we think the sentence will look like $\forall x(A)$. What comes next? Well, we are trying to say that for *any* person, there is *some* person such that the first loves the second. We have already written the quantifier that corresponds to 'any', so now we do the same for 'some'. Now our sentence looks like $\forall x \exists y(A)$.

Notice that we are using two different variables. This lets us know that we need to select different individuals to fill in each blank in the predicates, and also helps us keep track of which blank spaces correspond to which quantifiers. It is very important in translations to do this whenever there are multiple quanitifers.

All that's left now is to complete the relation. It should be easy to see that what we want is $\forall x \exists y(Lxy)$. Therefore, we have:

"Everyone loves someone" $= \forall x \exists y(Lxy)$

But now suppose that we want to say that there is a particular person such that everyone loves *that* person in particular. The way that we do that is by swapping quantifiers and giving the existential quantifier the wider scope. So we would say:

"There is someone that everyone loves" $= \exists y \forall x(Lxy)$

Notice that in this example we had to swap the order of the quantifiers, but we didn't have to swap the variables. It's customary with variable names to start with x, then y, then z, etc. but really you can pick them in any order. In this case I thought it was clearer to use this order because it preserved the relation 'Lxy' in both translations.

Let's try another example, this time throwing in a negation for good measure. We will translate the sentence "Someone does not love anyone". For this translation we see that the first quantifier relates to 'someone', i.e. we start with the existential quantifier. However, after that things get a bit trickier, because the second quantifier comes all the way at the end of the sentence. The easiest way to see what the correct translation will be is to notice that the word 'anyone' relates to the predicate 'loves'. Thus, the quantifier has to at least have scope over this predicate. Since the word is 'anyone', our first instinct should be the universal quantifier. All that's left after that is the negation, which has scope over the relation itself. That will produce the following translation:

"Someone does not love anyone" $= \exists x \forall y \neg(Lxy)$

Alternatively, we could use the logical equivalences we discussed earlier to shift the negation symbol relative to the quantifiers, which would give us $\exists x \neg \exists y(Lxy)$. We could even do this one more time, which would give $\neg \forall x \exists y(Lxy)$.

Do you notice anything interesting about this final version of the translation? It's the negation of the translation that we produced for "Everyone loves someone". Think about it! Isn't that just what we want? If it's not true that everyone loves someone, then there has to be at least one person that doesn't love anyone! And that's exactly what this sentence said!

These kinds of translations take a long time to master. One other thing we could do is mix in constant names with variables. For example, in the sentence "Someone has a crush on Becca, but Becca doesn't have a crush on anyone" there are both quantifiers and a definite name. In order to translate this sentence, then, we need to keep track of when to use variables and when to use a definite name. See if you can explain to yourself why this would be the correct translation:

$\exists x(Cxr) \wedge \neg\exists y(Cry)$ (Pay attention to the order of objects in the relations!)

For one last look at what's possible, take note of what happens to our translations when we throw extra predicates into the mix. For example, let's say that we are trying to translate the sentence "Every student enjoys at least one class." In this sentence, we have two quantifiers; 'every', and 'at least one'. However, the placement of the quantifiers in the sentence is important. Since 'every' comes first, we give it the widest scope. So we can start the sentence with '$\forall x$'. But what do we do after that? Well we still have a universal claim, which we know to translate as a conditional. Let's try to reword the original sentence in a way that makes this structure a bit more obvious. We might say it as

"If anything is a student, then there is something that is a class and the student enjoys it." This is a terrible way to say what we mean in English, but it helps us reveal the logical structure of the statement much more clearly. We can now see the conditional more clearly. Most importantly, we can see that the existential quantifier is actually *inside* of the conditional. So the way that we translate the sentence is

$\forall x(Sx \implies \exists y(Cy \wedge Exy))$

This trick of carefully rewording a complex sentence to better see where the quantifiers are in relation to one another is very helpful, and something that you can practice.

That about covers all of the kinds of translations into predicate logic that we'll be looking at in the course. Here are some more practice questions to end the discussion. After this, you should be ready to move on to Problem Set 7.

1. "Justin does not respect Victor."
2. "Justin respects someone."
3. "No one is loved by everyone."
4. "Dylan loves his parents."

5. "Some professors are liked by some students."

3.11 Answers to Practice Problems

Section 1.7

1. $\forall x(Fx \implies Ax) \wedge \neg\forall x(Ax \implies Fx)$
2. $\forall x(Sx \implies Px)$
3. $\exists x(Cx \wedge Rx)$
4. $\neg\exists x((Ex \wedge Sx) \wedge Px)$
5. $\forall x(Fx) \implies \exists x(Sx)$

Section 1.8

1. Sab
2. Lcd
3. Pef

Section 1.10

1. $\neg Rjv$
2. $\exists x(Rjx)$
3. $\neg\exists x\forall y(Lyx)$
4. $\forall x(Pxd \implies Ldx)$
5. $\exists x(Sx \wedge \exists y(Py \wedge Lxy))$

4 Proofs in Predicate Logic

Now that we understand how to translate English sentences into predicate logic, we can move on doing proofs in predicate logic as well. As our first example, we'll continue to look at our original syllogism about Socrates:

1. All humans are mortal.
2. Socrates is a human.

$\therefore$ 3. Socrates is mortal.

In this section, we will develop *rules of inference* for predicate logic that allow us to prove that this argument is valid. This process is identical to doing proofs in propositional/sentential logic, only with new rules added in to cover predicate logic translations.

4.1 Rules of Inference: Universal Instantiation

The first thing to keep in mind is that **every inference rule for sentential logic is also a rule of inference for predicate logic**. What this means is that everything that's on our inference rulesheet for **Unit 2** will also be on our rulesheet for doing predicate logic proofs. But we are also going to add on a few new rules, to cover both universal and existential statements.

Universal Instantiation (UI)

1. $\forall x(Fx)$

---

2. Fn (*where n is any name*)

The rule of **Universal Instantiation** tells us that when we have a universal statement, we can replace every occurrence of the variable associated with that universal statement with a specific name instead. This description is a mouthful, but it's actually quite intuitive. All it says is that if something is true about *everything*, then it's true about *each thing in particular.*

The most important insight about applying **UI** is that it does not matter which name we choose, since universal statements are true for *everything* in the domain of discourse. Therefore, we can choose literally any constant in the domain. Let's return to our original argument about Socrates and see how rule **UI** gives us the tools to prove that the argument is valid.

First, we will start off our proof as normal by listing out the premises at the top.

1	$\forall x(Hx \implies Mx)$	Pr
2	Hs	Pr
3	$Hs \implies Ms$	UI, 1
4	Ms	MP, 2, 3

The crucial step in this proof is the application of rule **UI** to line 1, which then allows us to use **MP** and arrive at the conclusion. Let us analyze how **UI** was applied in this particular case. First, we began with the universal statement "$\forall x(Hx \implies Mx)$". In order to apply **UI**, we first replace every instance of the variable x that is *inside* the scope of the universal quantifier. That covers both the variables after the predicate letters H and M. What do we replace this variable with? Well, we could choose any name we want, since the original sentence is a universal one. However, this particular argument is about Socrates, and Socrates is referred to by the constant 's' in line 2. Therefore, we would choose to replace the variable with the constant s. Once the variables are all replaced, we remove the universal quantifier on the outside of the sentence, which leaves us with line 3.

Using this new rule will take a bit of practice, and there are some important things to keep in mind. For instance, we must still obey the rule that we learned in **Unit 2**, that rules of inference must be applied to entire sentences, not just parts of sentences. We can see how this would come into play if we try to apply **UI** to the following sentence:

$\forall x(Ax) \implies \forall x(Bx)$

It is *not* possible to use **UI** on this sentence. The reason for this is that this sentence is not a universal statement. Its main connective is a conditional. Therefore, we can only apply the rules of inference pertaining to conditionals to it, not the rules for universal statements. Our quantifier rules of inference are extensions of the rules for sentential logic – **the fundamental rules of how to apply them precisely are the exact same**.

4.2 Quantifier Scope and Applying Quantifier Rules

You may have been somewhat confused by the use of **UI** on line 1 of the proof in the preceding section. That may have been because the rule (as stated) shows '$\forall x(Fx)'$, whereas line 1 of this proof shows '$\forall x(Hx \implies Mx)$'. This raises a very important point about the quantifier rules, which is that they apply to *every occurrence of the variable that appears under their immediate scope.* What does that mean? Well, when we consider '$\forall x(Hx \implies Mx)$', we see that the 'x' in 'Hx' and the 'x' in 'Mx' fall under the scope of the quantifier '$\forall x$'. Therefore, when applying **UI** to this statement, we need to replace both values of 'x' with a chosen name.

N.B. When applying a quantifier rule to a sentence, one must replace every instance of the quantified variable with the same name.

This also holds true for all of the other quantifier rules that we will see later on in the unit. For a bit of practice, try applying **UI** to each of the following sentences. The answers can be found at the end of the unit notes.

1. $\forall x(Fx \implies (Gx \vee Hx)$
2. $\forall x(\exists y(Fy) \implies Gx)$

3. $\forall y(Hy \implies \forall z(Fz \implies Gz))$

4.2.1 Practice: Basic Proofs Using Universal Instantiation

Each of the following basic proofs require the use of **UI**. Give them a try! Solutions are at the end of the unit.

(1) 1. $\forall x(Ax \wedge Bx)$

2. Cd

$\therefore$ 3. $(Ad \wedge Bd) \wedge Cd$

(2) 1. Aa

2. $\forall y(By \implies \neg Ay)$

$\therefore$ 3. $\neg Ba$

(3) 1. $\forall x(Ax) \implies \forall y(By)$

2. $\forall x(Ax)$

$\therefore$ 3. Ba

4.3 Rules of Inference: Existential Generalization

Rule **UI** shows us how we can deduce statements about particular members of the domain from universal statements. There is an analogous rule for existential statements as well, although it is a bit trickier to work around. In fact, the instantiation rule for existential statements is probably the most common cause of invalid predicate logic proofs. We will learn the rule shortly, and see how we can avoid the common pitfalls, but before that we might take a look at the more intutive and easier to use rule for existential statements: existential generalization.

Existential Generalization (EG)

1. Fn (*where* n *is any name*)

---

2. $\exists x(Fx)$

Rule **EG** tells us something very intuitive about members of a domain of discourse: if a particular member has a certain property, then there exists a member with that property. In fact, the rule is so obvious that it might even sound a bit silly to point it out. You might be thinking that there must be some hidden trick to this, or that there's more to it than first appears, but there isn't. Here are some English examples that show how simple the rule truly is.

E.g. 1

1. Rover is a dog.

$\therefore$ 2. A dog exists.

E.g. 2

1. Rover is a brown dog.

$\therefore$ 2. A brown dog exists (or: "something that is both brown and a dog exists").

This generalization rule is useful for doing proofs in predicate logic specifically because they are general statements about a domain of discourse as a whole. Let's use a short example to make this clear.

We will prove that the following argument is valid:

1. $\exists x(Fx) \implies \exists y(Gy)$
2. Fa

$\therefore$ 3. $\exists y(Gy)$

1	$\exists x(Fx) \implies \exists y(Gy)$	Pr
2	Fa	Pr
3	$\exists x(Fx)$	EG, 2
4	$\exists y(Gy)$	MP, 1, 3

4.3.1 Practice: Basic Proofs Using Existential Generalization

Each of the following proofs requires the use of **EG** (some of them use **UI** as well!). Try them out as practice. There are solutions at the end of the chapter.

(1) 1. Fa

2. Ga

3. $(\exists x(Fx) \wedge \exists x(Gx)) \implies \exists x(Hx)$

$\therefore$ 4. $\exists x(Hx)$

(2) 1. $\forall x(Fx)$

$\therefore$ 2. $\exists x(Fx)$

4.4 Rules of Inference: Existential Instantiation

You may have noticed that we've looked at two different kinds of rules so far: first, we looked at an *instantiation* rule, and then at a *generalization* rule. The first kind of rule – instantiation – taught us how to take a general, universal statement, and derive a statement about a particular object in the domain. On the other hand, the generalization rule taught us how to take a statement about a particular object and derive a more general claim about the entire domain.

One natural thought that you might be having is that there must also be a generalization rule for universal statements, and an instantiation rule for existential statements. Well, you'd be right to think that, because there are such rules! However, the key difference between those two rules and the ones we've already discussed is that they have very restricted uses, and extra rules for when it's possible to use them. That's why we saved them for last.

In this subsection, we will focus on understanding the existential instantiation rule and the pitfalls associated with it that need to be avoided. Let's look at the rule itself.

Existential Instantiation (EI)

1. $\exists x(Fx)$

---

2. Fn (*where n is any* **new** *name*)

The most important thing about using rule **EI** is that the name that you choose must be a new name – i.e. **it must not appear in the proof on any line before the application of the rule**. We will look at an example shortly to explain more what this means, but we should also talk through it a bit to make sense of why this restriction is in place.

Consider what an existential statement like "$\exists x(Fx)$" means. This statements says "There is something that has the property of being 'F"'. What the statement does not tell us is *which* thing has that property. That's why we need to choose a new name whenever we use this rule in the context of a proof. Whenever we use rule **EI**, we are basically saying something like "something has this property, but I don't know anything else about that thing. So I'm going to pick a new name that means 'the thing that has this property', because that's all I know about it for certain."

Not remembering to pick a new name is one of the most common ways that predicate logic proofs get messed up. This happens especially often when existential and universal statements appear in the same proof. It will take a lot of practice to remember the correct order.

As usual with these more abstract restrictions, it can be difficult to understand exactly what they're saying without a concrete example. Let's look at an invalid argument to see how the incorrect use of **EI** can run us into difficulty.

1. $\exists x(Fx)$
2. $\exists x(Gx)$

$\therefore$ 3. $\exists x(Fx \wedge Gx)$

As I mentioned, this argument is invalid. However, if we're not careful about the restriction on using **EI**, we might fool ourselves into thinking that the following 'proof' shows that it is valid.

1	$\exists x(Fx)$	Pr
2	$\exists x(Gx)$	Pr
3	Fa	EI, 1
4	Ga	EI, 2
5	$Fa \wedge Ga$	Conj, 3, 4
6	$\exists x(Fx \wedge Gx)$	EG, 5

You may have already noticed the error – line 4 is an incorrect application of rule **EI**. However, you might not be convinced that the restriction makes sense. Maybe the proof makes the argument seem intuitively valid to you after all. We can give an English example of this sort of argument to show definitively that the argument form is invalid. Consider the following:

1. Some numbers are even.
2. Some numbers are odd.

$\therefore$ 3. Some numbers are both even and odd.

This argument is an instance of the original. In it, we restrict the domain of discourse to just numbers. The first premise says that an even number exists. The second says that an odd number exists. The conclusion asserts that some number that is both even and odd exists as well. However, we know that this has to be false! There is no such thing as an even and odd number. In fact, we can generate a practically infinite number of counterexamples to this basic argument form, just by picking any two, mutually exclusive properties that some objects have (being round and being square, being over six feet tall and being less than six feet tall, and so on).

This simple English argument shows us very clearly why the restriction on **EI** is necessary for it to work as a deductive rule. Here are a few tips to avoiding getting into difficulty when trying to do proofs that require the use of the rule:

a. Use notes in the margin of the page to remember when you choose a new name using **EI**.
b. Remember that if a specific name appears in the conclusion of a proof, then it counts as being 'used' for the purpose of applying **EI**.
c. Always apply **EI** as many times as necessary *before any* applications of **UI**. This doesn't make a practical difference to how your proof functions, but it will help you to keep the restrictions in mind for the times when it is crucial to the proof.

4.4.1 Practice: Basic Proofs Using Existential Instantiation

Finally, here are a couple of practice proofs that involve using **EI**.

(1) 1. $\exists x(Fx)$
2. $\forall x(Fx \implies Gx)$
$\therefore$ 3. $\exists x(Gx)$

(2) 1. $\exists x(Fx \wedge Gx)$
2. $\exists x(Gx) \implies \exists x(Hx)$
$\therefore$ 3. $\exists x(Hx)$

4.5 Rules of Inference: Universal Generalization

There is only one more rule to discuss, and that's universal generalization. As was mentioned in the previous section, this rule is only possible in some restricted cases, just like existential instantiation. Before we get to the rule itself, let's consider a problem where it seems intuitively obvious that we should be able to make a universal generalization, to better understand what the rule is supposed to accomplish.

1. $\forall x(Fx \implies Gx)$
2. $\forall x(Fx)$

$\therefore$ 3. $\forall x(Gx)$

This is a valid argument. However, none of our rules so far allow us to prove that! What we can do, though, is prove 'Gn' for any name, 'n', like this:

1	$\forall x(Fx \implies Gx)$	Pr
2	$\forall x(Fx)$	Pr
3	$Fa \implies Ga$	UI, 1
4	Fa	UI, 2
5	Ga	MP, 3, 4

1	$\forall x(Fx \implies Gx)$	Pr
2	$\forall x(Fx)$	Pr
3	$Fb \implies Gb$	UI, 1
4	Fb	UI, 2
5	Gb	MP, 3, 4

1	$\forall x(Fx \implies Gx)$	Pr
2	$\forall x(Fx)$	Pr
3	$Fc \implies Gc$	UI, 1
4	Fc	UI, 2
5	Gc	MP, 3, 4

As you can see, we can repeat this process over and over using a different name each time, until we run out of names. But if that's possible, then isn't that the same thing as saying that $\forall x(Gx)$? Well, yes it is! So in this case, it's pretty clear that $\forall x(Gx)$ does actually follow from the two premises that we were given. Therefore, the question is how we can develop a valid rule of inference to represent this fact, and get us to the conclusion without needing to prove that it's true for each name individually.

The key to doing this is to realize that, in each instance of this argument, the name we picked was *arbitrary*. Each time, we could have chosen any name whatsoever and the proof would have remained exactly the same. That's how we know that this proof can be generalized: the specific name we chose did not matter! Why didn't it matter? Because it doesn't show up anywhere else in the proof, besides the places where we used a universal instantiation. This allows us to formulate the rule as follows:

Universal Generalization (UG)

$$\begin{array}{l} 1.\ Fn \\ \hline 2.\ \forall x(Fx)\ \ ** \end{array}$$

**

1. Where '$\forall x(Fx)$' is obtained by replacing every occurrence of 'n' in 'Fn' with 'x', and
2. The name 'n' does not occur in any premise, assumption, or earlier line that is obtained using rule **EI**.

Here is the way that our earlier proof would look, now that we have formalized rule **UG**.

1	$\forall x(Fx \implies Gx)$	Pr
2	$\forall x(Fx)$	Pr
3	$Fa \implies Ga$	UI, 1
4	Fa	UI, 2
5	Ga	MP, 3, 4
6	$\forall x(Gx)$	UG, 5

In this case, we know that we can use **UG** because 'a' does not appear in any of the premises of the argument, and it also does not appear in any line that results from an application of rule **EI**. That means that the name 'a' was chosen arbitrarily and could have been replaced by any other name, allowing us to generalize the result on line 5 universally. Let's look at a slightly more complicated example, this time involving a proof where some names do occur early on in the premises.

1. $\exists x(Fx \wedge Gx)$
2. $\forall x(Fx \implies \forall y(Gy))$
3. $\forall x(Gx \implies Fx)$

$\therefore$ 4. $\forall x(Fx)$

The proof looks like this:

1	$\exists(Fx \wedge Gx)$	Pr
2	$\forall x(Fx \implies \forall y(Gy)$	Pr
3	$\forall x(Gx \implies Fx)$	Pr
4	$Fa \wedge Ga$	EI, 1
5	Fa	Simp, 4
6	$Fa \implies \forall y(Gy)$	UI, 2
7	$\forall y(Gy)$	MP, 5, 6
8	Gb	UI, 7
9	$Gb \implies Fb$	UI, 3
10	Fb	MP, 8, 9
11	$\forall x(Fx)$	UG, 10

Now we can go through the proof step by step to make sure that we have done everything correctly. The most important lines to pay attention to are steps lines 4 and 8. On line 4, we use **EI**. As per the restrictions for that rule, we must pick a new name that does not appear anywhere else in the proof. In this case, we go with 'a'.

When we get to line 8, we have to make another choice of name. Let's pretend that we chose 'a' again. That's allowed at this stage, since we are using **UI** – we can pick any name we want. Notice, however, how that alters the proof. We would have to change line 9 as well, and that would give us 'Fa' on line 10. That doesn't help us at all! We already have that on line 5. We're trying to prove something universal, which makes our choice of name absolutely critical. We have to pick a name *besides* 'a', since 'a' appears earlier in the proof as the result of using **EI**, which the restrictions on **UG** forbids. Instead, we pick a new name on line 8. It can be any name we want except 'a'. Here, we chose 'b'. But since we could have picked any name besides 'a', what we have done is effectively prove that everything is 'F', since any name besides 'a' would work, and line 5 already says that a is F. This proof adheres to all of the restrictions for using **UG**, so we can use it on line 11 to get the conclusion.

Finally, let's look at an example where **UG** cannot be used, i.e. is an invalid deduction.

1. $Ga \wedge Gb$

$\therefore$ 2. $\forall x(Gx)$

This argument is invalid! An *incorrect* proof using **UG** would look like this:

1	$Ga \wedge Fa$	Pr
2	Ga	Simp, 1
3	$\forall x(Gx)$	A WRONG AND NOT ALLOWED USE of UG, 2

This is invalid. From the definition of **UG**, we can see that line 3 violates the restrictions. That is because 'Ga' appears earlier on in the proof, as part of a premise. Another way of looking at things is that we would need to use the specific name 'a' as it appears in line 1 in order to arrive at line 2, that is we could not get line 2 to use any arbitrary name, it would need to be the specific name 'a'. **UG** only works when the same proof could be done by swapping out the particular name for any other name without changing the rest of the proof. This is not true here since, for instance, there would be no way to get a line that says 'Gb' or 'Gc'. Hence, **UG** cannot be applied.

4.6 A More Complex Predicate Logic Proof

In the final section of the notes, we will take a look at a more complicated proof that combines our new predicate logic rules with the rules of Unit 2. In this complicated proof, we will take care in applying all of the rules correctly. We will prove one of the two quantifier negation rules that we saw earlier in the chapter (pg. 12). This is a good example of a predicate logic proof that uses a wide range of techniques from across both Units 2 and 3. The second proof will be left as an exercise on Problem Set 8.

Prove: $\neg\forall x(Fx) \iff \exists x\neg(Fx)$

1	$\neg\forall x(Fx)$	As
2	$\neg\exists x\neg(Fx)$	As
3	$\neg Fa$	As
4	$\exists x\neg(Fx)$	EG, 3
5	$\perp$	2,4
6	Fa	IP, 3–5
7	$\forall x(Fx)$	UG, 6
8	$\perp$	1,7
9	$\exists x\neg(Fx)$	IP, 2–8
10	$\neg\forall x(Fx) \implies \exists x\neg(Fx)$	CP, 1–9
11	$\exists x\neg(Fx)$	As
12	$\forall x(Fx)$	As
13	$\neg Fb$	EI, 11
14	Fb	UI, 12
15	$\perp$	13,14
16	$\neg\forall x(Fx)$	IP, 12–15
17	$\exists x\neg(Fx) \implies \neg\forall x(Fx)$	CP, 11–16
18	$\neg\forall x(Fx) \iff \exists x\neg(Fx)$	Eqiv, 10, 17

It is important to note that this proof would work no matter how complex the formula after the quantifier is – these are general rules that hold for all quantified statements.

Try going through this proof one line at a time to see what each step is doing. It may be difficult to understand because of unfamiliarity with these new rules, but it is possible if you take it slow and keep track of the assumptions.

Proofs like these represent the culmination of a lot of hard work on your part throughout the entire course. Even after this, there is a lot to cover. For example, it is possible to do predicate logic proofs with relational predicates as well. After that, we could develop the notion of 'identity' – when two things are identical to each other, which allows us to do even more.[40] First order logic

[40] Without a concept like identity, it is not possible to translate English sentences like "There are at least two people on Earth" into predicate logic.

is a big topic. At the end of the day, this was an introductory course, but we covered a lot! You should feel very proud of yourself for pushing through this much material! Congratulations!

5 Solutions to Predicate Logic Practice Problems

5.1 Universal Instantiation Practice (2.1.1)

1. $Fa \implies (Ga \lor Ha)$
2. $\exists y(Fy) \implies Gb$
3. $Ha \implies \forall z(Fz \implies Gz)$

5.2 Basic Proofs Using Universal Instantiation (2.1.2)

(1)

1	$\forall x(Ax \land Bx)$	Pr
2	Cd	Pr
3	$Ad \land Bd$	UI, 1
4	$(Ad \land Bd) \land Cd$	Conj, 2, 3

(2)

1	Aa	Pr
2	$\forall y(By \implies \neg Ay)$	Pr
3	$\neg\neg Aa$	DN, 1
4	$Ba \implies \neg Aa$	UI, 2
5	$\neg Ba$	MT, 3, 4

(3)

1	$\forall x(Ax) \implies \forall y(By)$	Pr
2	$\forall x(Ax)$	Pr
3	$\forall y(By)$	MP, 1, 2
4	Ba	UI, 3

5.3 Basic Proofs Using Existential Generalization (2.2.1)

(1)

1	Fa	Pr
2	Ga	Pr
3	$(\exists x(Fx) \wedge \exists x(Gx)) \implies \exists x(Hx)$	Pr
4	$\exists x(Fx)$	EG, 1
5	$\exists x(Gx)$	EG, 2
6	$\exists x(Fx) \wedge \exists x(Gx)$	Conj, 4, 5
7	$\exists x(Hx)$	MP, 3, 6

(2)

1	$\forall x(Fx)$	Pr
2	Fa	UI, 1
3	$\exists x(Fx)$	EG, 2

5.4 Basic Proofs Using Existential Instantiation (2.3.1)

(1)

1	$\exists x(Fx)$	Pr
2	$\forall x(Fx \implies Gx)$	Pr
3	Fa	EI, 1
4	$Fa \implies Ga$	UI, 2
5	Ga	MP, 3, 4
6	$\exists x(Gx)$	EG, 5

(2)

1	$\exists x(Fx \wedge Gx)$	Pr
2	$\exists x(Gx) \implies \exists x(Hx)$	Pr
3	$Fa \wedge Ga$	EI, 1
4	Ga	Simp, 3
5	$\exists x(Gx)$	EG, 4
6	$\exists x(Hx)$	MP, 2, 5