## Logic, Critical Thinking, and Computer Science

### Logic: A Brief Introduction

This course will teach you the fundamentals of formal logic and show you how logical thinking and logical concepts were applied to design and build digital machines. By gaining skills in logical analysis, you will not only gain important life skills, but you'll better understand how computers work which will make you a better technician in the long run.

As we'll see more later, logic is a science that sets forth rules for properly-ordered thinking while critical thinking mainly is the application of those rules. Many introductory students tend to think that logic is imposed on human thinking. It's better to think of logic as something philosophers and mathematicians have discovered about human thinking. So logical rules are like laws of nature: they were formed out of years of observation on how thinking works when it's working well.

### The Wheres and Whys of Logic and Critical Thinking

Although nobody has discovered a handy formula that sums it all up, many individuals down through the centuries have thought deeply about thinking, and a number of standards, or criteria, have been proposed, tested, and found to be reliable guides to sound judgment and the attainment of truth, or correspondence with reality. For an example of such a criterion, consider statistical thinking. To learn about a large population of things, we sometimes examine a sample and then conclude something about the population as a whole on the basis of what we observe in the sample. It is common sense that the larger the sample, in relation to the whole population, the more likely the conclusion is to be true. It is also common sense that the more randomly selected the sample, the more likely the conclusion is to be true.

Thus, critical thinking is more than mere criterial thinking. Rather, it is thinking on the basis of criteria that have been tested and found to be reliable guides to sound judgment and the attainment of the truth about the matter under consideration. Someone who makes every important decision on the basis of "throwing the bones" is engaged in criterial thinking: throwing the bones is this person’s criterion. (Throwing the bones is an ancient form of divination in which animal bones are tossed onto a mat and the pattern is interpreted, usually by a shaman, spiritual reader, or fortune-teller of some sort.) However, such a person is not a critical thinker -- at least as the words critical thinker are used today -- for his or her criterion has not been tested and found to be a generally reliable guide to truth and sound judgment.

### Wax on, Wax off

As you work through the course, you may find yourself asking, "I thought this was a computer science course so why am I learning all this logic that has nothing to do with computers?" If you've seen the 1984 movie The Karate Kid, you may remember that young Daniel asked Mr. Miyagi to teach him Karate. On the first few days Daniel showed up for training Mr. Miyagi "assigned" Daniel a bunch of chores around Miyagi's house like painting his fence, sanding his deck, and waxing his cars. After three days of back-breaking work, Daniel lashes out in rage at Mr. Miyagi complaining that he hasn't learned any Karate. Mr. Miyagi replies, "You learn plenty." Daniel retorts, "I learn plenty, yeah, I learned how to sand your decks maybe. I washed your car, paint your house, paint your fence. I learn plenty!" Patiently Mr. Miyagi teaches Daniel, "Ah, not everything is as it seems. . ." and then shows Daniel how the muscle memory he was building from doing the chores had to be learned first before he could actually develop the skills to do Karate properly.

While many of the lessons, exercises, and assessments will not seem to directly relate to computer programming or computer science, you will be learning something important: how to think logically. You will be developing very important muscles that will serve you immensely when you learn how to program. If you already program, the skills you learn in this course will help you become a better programmer.

So be patient and work carefully through the lessons. When you've mastered the content, your mind will have gained important skills that you can not only apply to work you do in computer science but will serve you in all of your life.

### Where do computers come in?

One of the primary ideas this course will explore is the relationship between logical and critical thinking and computer science. In order to establish that relationship, let's first take a look at how computers work and then we'll dive into basic principles of formal logic.

## What is a Turing Machine?

In order to better understand how computers "think," we can start with the simplest version of a computer: the Turing machine. A Turing machine isn't really a machine but a model for describing how computers do what they do.

The model was invented by computer scientist Alan Turing who many consider the father of digital computing. Turing realized that all computational problems can be broken down into a very simple language, a digital language: 0 and 1. That's pretty simple, right? One way to think about Turing's insight is in terms of two states like "on" and "off", "true" and "false, "in" and "out" or "yes" and "no". You can create a Turing machine out of anything that can be in two different states. One philosopher suggested that a Turing machine can be made from birds pecking: when the bird's head is up, it's in the "0" position and when it's down, it's in the "1" position.

In many ways, the light switch on your wall is a digital machine. When the light switch is on, it's in one state and when it's off it's in another--and those are the only two states the light switch can be in. While not a lot of information can be communicated with that switch, just by looking at it, we can tell what state the light is in (assume for a second that you can't see the actual light the switch controls).

All modern computers are essentially very complex Turing machines turning switches--lots of switches--on and off very rapidly to do all the magical things a computer does.

## But Wait!

How can a switch create all the interesting graphics, process data in a spreadsheet, create text messages and digital phone calls, produce holographic images, and all the other cool things our computers do? It's a bit more complicated than a simple light switch (you probably knew this was coming).

The foundation of computer systems and the logic that makes them up are made up of sets of switches all working together. Let's extend our light switch example to see how this works. Suppose we have two light switches that when used in various combinations produce different colored lights. Here are the combinations we can create (using 0 for off and 1 for on):

| **Switch 1** | **Switch 2** |
| --- | --- |
| 0 | 0 |
| 0 | 1 |
| 1 | 0 |
| 1 | 1 |

If each combination "turns on" a different color, we can now have three different color lights using just two switches (when both switches are off the light is off). For example:

| **Switch 1** | **Switch 2** | **Color** |
| --- | --- | --- |
| 0 | 0 | Off |
| 0 | 1 | Red |
| 1 | 0 | Green |
| 1 | 1 | Blue |

If we add just one more switch, we can add four more colors! This is the insight that Alan Turing discovered. By using this simple digital language you can create very complex combinations that can represent just about anything.

## Bits and Bytes

In the previous lesson we saw how a simple digital language of 0 and 1 provides the foundation for a type of language that we can use to do work. In this lesson, we'll explore how that simple language was expanded to create the foundation for the computers we use today.

The basic Turing machine uses 0 and 1 to create a simple "on" and "off" model. In computer science, this is called a "binary" model. The word "binary" comes from the Latin word, "binarius" which means two together or a pair. So 0 and 1 are a pair of digits or a "binary digit." This phrase was shortened to the word "bit." A bit in computer science then is the two states 0 and 1 and the basic unit of the binary system computers use. Like a light switch, a bit can only be either 0 or 1 at any given time but has the potential to be either at a given time.

Computer systems then are built from this simple model of bits in combination with other bits. In the last lesson we considered two light switches that, when combined in different states, produces different colored lights:

| **Switch 1** | **Switch 2** | **Color** |
| --- | --- | --- |
| 0 | 0 | Off |
| 0 | 1 | Red |
| 1 | 0 | Green |
| 1 | 1 | Blue |

In the language of bits, this a two bit system. We have two binary states that work together to form four different combinations producing a lights off state and three different colored lights on states. Modern computers use eight bits in combination to form the fundamental unit of digital information. This unit is called a **byte**.

In the very first modern computers a byte with its bits in various combinations was used to represent a single character of text. The table below shows the byte representation of the letter "A" in binary language:

| **Bit 1** | **Bit 2** | **Bit 3** | **Bit 4** | **Bit 5** | **Bit 6** | **Bit 7** | **Bit 8** | **Letter** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | A |

When a user presses the letter A on her computer, the keyboard sends a signal to the processor of the computer that translates the signal into the byte representation above. When the CPU (central processsing unit -- the "brains" of the computer") gets that digital combination, it knows that it needs to send another signal to the screen to draw the letter "A". All modern computers at their core use a similar model to create the complex operations that you enjoy on your phone, tablet, or desktop computer.

## Algorithms

We have a very basic framework for how modern computers work. This basic model of bits and bytes form the foundation for the operations of a computer system that do the real work. The binary model we studied earlier is the fundamental language the microprocessors in your computer understands. At a much higher level, humans write code in programming languages that computers compile to something they can process. It is this programming code that we now need to look at in order to understand how logic plays a role.

Computers are particularly useful because they perform repeatable tasks in a predictable way (even when you're frustrated with your computer for not doing what you want). When you open a word processor or play a computer game, the computer operates against a set of commands or instructions that a team of programmers wrote and those instructions tell the computer what to do and when it needs to do them. These instructions always do the same thing and, when the program is written well, it does them consistently and without errors.

## Algorithms in Real Life

These instructions are called "algorithms". Author Yuval Noah Harari defines an algorithm this way: "An algorithm is a methodical set of steps that can be used to make calculations, resolve problems and reach decisions. An algorithm isn’t a particular calculation, but the method followed when making the calculation." Some have referred to an algorithm as a recipe. But put simply, **an algorithm is a repeatable set of steps that can take inputs and that produce a predictable, consistent output.**

There are many popular metaphors that educators have used to illustrate algorithms and one of my favorites is the steps used when making a peanut butter and jelly sandwich. We can describes the steps--the algorithm--for making this sandwich this way:

Step 1: set out the ingredients: peanut butter, jelly, bread, and a spreader

Step 2: open the jars of peanut butter and jelly

Step 3: Set out two pieces of bread on a flat surface

Step 4: Spread a thin amount of peanut butter on one slice of bread

Step 5: Spread a thin amount of jelly on the other slice of bread

Step 6: Place one slice of bread on top of the other

Step 7: Enjoy!

Notice that this is a very rudimentary description. We can get much more specific on each of the steps. For example, we could specify exactly how much peanut butter and jelly to spread in steps 4 and 5. In step 6, we didn't specify that the bread should be placed on top of each other with the peanut butter and jelly sides of the bread facing each other (not doing this would result in a very messy sandwich). But the point is, we've described a process for making the sandwich and the process is repeatable and always results in the same output: a sandwich.

Of course, because this is a very imprecise set of instructions, each sandwich will turn out a little bit different. Still, at a very high level, we end up with essentially the same results. Computers follow similar instructions but computer systems are very precise and the instructions they follow generally result in output that is much more consistent.

## Algorithms and Computer Science

When programmers write computer programs, they essentially are writing recipes just like this to tell the computer what to do. Programs are written in sets of routines that specify what the computer should do in various circumstances. For example, you could write a computer program that adds two numbers together. The computer user might specify the numbers he or she wants to add, and your algorithm will do the addition operation and output the result. You would write your program in a specific language like JavaScript or C# (pronounced C-sharp) and that language--which makes it easy for humans to use--is compiled to the bits and bytes we talked about earlier so the computer can understand it. But it's still an algorithm at the end of the day.

## Algorithms and Logic

We're now getting closer to understanding the relationship between computer science and logic. As we'll see in the next lesson, logic follows this algorithmic model by describing a consistent way in which ideas should relate to one another. It provides a set of recipes we can use to organize thought. We'll look more closely at this idea in the next lesson.

## Logic and Computer Science

Now that we have a basic understanding of how computers work and the relationship of that model to algorithms, we can begin to look at how computer science and the system of formal logic used by humans every day (as well as in formal disciplines like philosophy, law, and science).

Logic is a science that analyzes concepts and ideas and attempts to provide rules for ordered thinking and find fallacies for improper thinking. Computers use programs to process data and produce some kind of output like images, music, spreadsheets, and online courses. So how are these two things related?

## Consider the following

Here's a simple computer program (written in pseudo code -- not a real programming language but a teaching device we can use to resemble one):

for (x=0; x < 7; x++){

If x is Even Then

print "Eat pasta for dinner"

Else

print "Eat salad for dinner"

}

This program routine is what is called a "for loop" and will increment the value of the letter x by 1 each time through the loop. To use the language of the previous lesson, this is an algorithm.

You can think of how this particular algorithm functions like going around a merry-go-round 6 times carrying a bucket that starts out with a single coin in it. Each time you pass a certain point, a coin is added to the bucket until you have 7 coins in your bucket. During the rest of the trip around the merry-go-round, you're doing something interesting like taking pictures, waving at friends and family, and eating popcorn. When you have the 7th coin, the merry-go-round takes one more trip around and stops before you get the 8th coin.

In this particular program, the for loop will check the value of the variable x which changes (it's incremented by 1) each time through the loop. If the value of x on its trip around the loop is even, then the program will print the sentence "Eat pasta for dinner". If the value of x is anything other than even--the "else" condition--then the program will print the sentence "Eat salad for dinner". Since the number assigned to x is a whole number, it can only be odd or even so we know the else condition for x will always be odd. We just made our very first program: a very rudimentary meal planner!

## From programs to logic

We'll look more closely at exactly what logic is in the next lesson. But let's start to explore how logic functions "algorithmically" by briefly looking at the relationship between what a computer program does and how it relates to a logical argument.

We can translate this computer program into an argument of formal logic very easily. Instead of a for loop, we'll use the physical calendar as the "iterator"-- the real-world object that will change the value we care about. In other words, the days of the week become the x of our argument and as each day passes, we have a new value for x. We now can write a logical argument that tells us what to eat on a given day.

For the example, we'll start with a deductive syllogism called a disjunctive syllogism. We'll learn more about this later on in the course. But put simply, the disjunctive syllogism takes any two values and tells us that when given those two values, if its not the first value, it must be the second. We can write our syllogism this way:

Premise 1: Either the day is odd or the day is even

Premise 2: it's not the case that the day is even

Conclusion: Therefore, the day is odd

This is a good start but this argument doesn't tell us what we will eat on each day. So we need to add another syllogism called a modus ponens to help us with that problem. This syllogism for our argument looks like this:

Premise 1: If the day is even, then we eat pasta

Premise 2: The day is even

Conclusion: Therefore, we eat pasta

Of course we need another syllogism for the odd days:

Premise 1: If the day is odd, then we eat salad

Premise 2: The day is odd

Conclusion: Therefore, we eat salad

We can now combine these into a single argument:

Given: The current day of the week

Premise 1: Either the day is odd or the day is even

Premise 2: If the day is even, then we eat pasta

Premise 3: If the day is odd, then we eat salad

Premise 4: It is not the case that the day is [odd/even]

Premise 5: The day is [even/odd]

Conclusion: Therefore we eat [pasta/salad]

Because our disjunctive syllogism rule says that if one of the options isn't true, the other must be true, we can combine the days of the week in premises 4 and 5 and the meal we eat in the conclusion and let the day of the week determine which we choose.

You'll notice that the computer program routine is much simpler. But the point we're illustrating is that the computer program can be translated into standard logical form and vice versa. We'll see why this works as we move throughout the course but you can see the tight relationship between computer logic and formal logic. If you can learn to apply a logical framework to your thinking in everyday life, it will help you think through how to write better computer programs--and vice versa! Apple founder Steve Jobs has been quoted as saying, "I think everyone should learn how to program a computer, because it teaches you how to think." Now we can start to see why!

## An Introduction to Logic

In this lesson, we'll look at what logic is and the basic structure of logical analysis. In subsequent modules, we'll dig deeper into how to construct and analyze arguments and apply logical arguments to critical thinking.

## What is Logic?

Put simply, formal logic is a system of the rules of right reasoning. Logic helps us determine when an argument has the right form and when it doesn't. Sometimes when students hear the phrase, "right reasoning" they complain, "But who gets to determine what is right and wrong? Wasn't logic just invented by some old, dead guys? Why do they get to tell us what is right and wrong in the way we think?" These are very good questions!

**First**, by using the term "right," logicians don't mean that logic tells us what to think. Logic doesn't tell you to think like a conservative or progressive, or like a socialist or a capitalist, or even an Apple fan or Microsoft fan. These ideas make up the content of your thoughts. Logic deals primarily with the structure of our thinking. Just like you can't make 2+2 equal to 47 no matter how hard you try, there are certain relationships ideas should be in relative to each other and logic helps us figure out what those relationships should be.

**Second**, like mathematics, logic wasn't invented but discovered. Logicians like Aristotle or Ada Lovelace examined relationships between ideas and terms and discovered which ones worked and which ones didn't and created rules for the relationships that worked. These rules became the system of formal logic.

### The First Logician

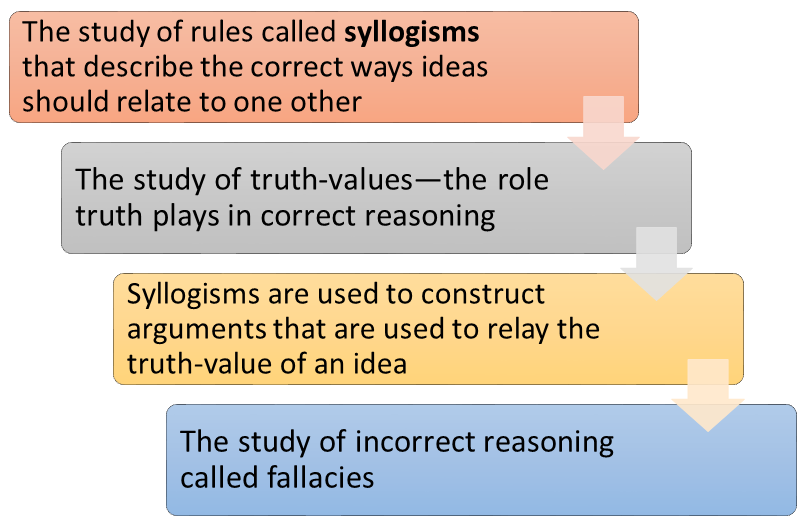
While its certainly true that humans were using logical thinking well before the Greeks and even formalized some of the rules, the first formal Western logician is widely considered to be Aristotle (died in 322 BCE). We'll see later that Aristotle's teacher Plato, and Plato's teacher Socrates also contributed to the system Aristotle developed. But Aristotle was the first person to systematize the rules that made up formal logic.

The "Big Three" Greek philosophers were Socrates, Plato, and Aristotle. Socrates taught Plato and Aristotle was Plato's student. One way to remember the order of these thinkers is to take the first letter of their names and put them in reverse alphabetical order: S comes last in the alphabet so Socrates comes first, P is in the middle, so Plato is next, and A comes first in the alphabet so Aristotle is last in the lineup.

### The Science of Logic

Logic, then, provides rules for properly ordered thinking and logical analysis is the science that applies those rules to our thinking. More specifically, logic helps us analyze arguments. We'll talk more about arguments in a bit but for now let's introduce the following terms in order to better describe what logic is.

1. Logic is the study of the rules of properly ordered thinking. The rules describe the correct ways ideas should relate to one another.
2. Logic also studies truth values -- the role truth plays in correct reasoning
3. The rules of logic make up arguments that are used to convey the truth value of ideas
4. Logic also studies incorrect reasoning called fallacies



## Arguments

# Introduction to Logic: Arguments

In this lesson, we'll be looking at the focus or object of logical analysis called arguments. You'll learn how arguments are structured and the components that make up arguments. In future lessons, you'll learn how to construct arguments and practice on your own.

## What is an argument?

In a previous lesson, we learned that logic is the science of properly ordered thinking. Every science has a focal point or object of its study and the object of logic is arguments. Put another way, arguments are the way we structure our thinking.

We use the word argument in every day usage to usually mean a verbal fight. If you get in an argument with a friend, co-worker or family member, it usually means that you have some disagreement and you're talking (or yelling) about your position and how it differs from the person you're arguing with.

In logic, an argument takes on a much more specific purpose and has a formal structure. Specifically, an argument is a structured set of sentences designed to convince someone of something. The structure is very important. It's so important in fact that logicians have given names to the parts of an argument. So here is a textbook definition of an argument:

An argument consists of: reasons (called **premises**) offered in support of a claim (called the **conclusion**).

When you first read that definition, you might see two components. But there actually are three and the third component is the most important. Let's look at each part more closely.

1. **Premises**: the reasons or evidence cited
2. **Conclusion**: the statement being proved by the premises
3. **A logical relation**: the logical connection from the premises to the conclusion

Remember that when you give an argument, the recipient of that argument should hear your premises and from the premises draw the conclusion. That is, the premises should cause the hearer to agree that your conclusion is true. That is why the logical relation is so important. Consider the following example:

Premise 1: Satya Nadella is the CEO of Microsoft  
Premise 2: The grass is the color green  
Conclusion: Therefore the sun will rise in the morning

Notice this appears to have the structure of an argument--we have two premises and a conclusion. However the premises in no way support the conclusion. They give us no reason to believe that the conclusion is true, even if we agreed that the premises were true.

Now consider this example:

Premise 1: All vegetables are plants  
Premise 2: This tomato is a vegetable  
Conclusion: Therefore this tomato is a plant

If you at least assume the two premises are true, then you can't help but agree with the conclusion. The conclusion is true based on the truth of the premises. This is a argument that fulfills the definition above. We have premises, a conclusion, and most importantly, a logical relationship between them.

You might ask, "What if I don't agree with the truth of the premises?" That's a good question and we'll talk about truth later in the course. But for now, we can add the qualifier, "assuming the premises are true" then the conclusion follows. When testing for a logical relation, you can assume the truth of the premises to see if the conclusion follows from them. But more on that later.

## Introduction to Logic: Statements

In the last lesson we saw that an argument is made up of premises, a conclusion, and a logical relation. We now need to look more closely at the premises and conclusion to see what types of things they are. For example, can we use any sentence as a premise? The answer is no, we can't! The premises and conclusion are very specific types of sentences.

While our examples in this course are in English, the principles apply to any language that makes declarative claims. We'll see why this is so shortly.

When we write (or speak) premises and conclusions, we are writing what are called **declarative statements** or just **statements**. Put simply, a statement is a language-based sentence that expresses a truth or possible truth:

* All US citizens live in the US
* Most businesses in Seattle use red carpeting in their offices
* Apple has a higher market capitalization than Microsoft

Notice that each of these statements attempt to declare a truth. Of course, the first two sentences are false and the third (as of this writing) is true. But for the purposes of logic, each qualifies as a statement because it is declaring something that may possibly be true. In the English language, these are called "declaratives." Only declaratives qualify as statements in logic. Here are various types of speech acts so you can compare and contrast them all.

1. Questions or interrogative: "What time is it?"
2. Commands or imperative: "Close the door!"
3. Exclamations or exclamatory: "Ouch!"
4. Performatives: "I promise" or "I thee wed"
5. **Declaratives**: "Carbon is an element"

Only declaratives qualify as statements because these are the only types of sentences that make a claim that something is true and this is what is important in logic.

## Introduction to Logic: More on Statements

In the last lesson, we saw that premises and conclusions are made up of statements--declartive sentences that make claim that something is possibly true.

## Simple and Compound Statements

A statement that declares a single truth is called a **simple statement**. In English, these are grammatically simple too. They typically have a subject, verb, and object:

* Ann is home
* Bob is home

You can see that each of these sentences declare that something is true and that only one situation or state of affairs is true. When writing logical arguments, it's good practice to break down your premises into simple statements as much as you can. This makes determining with the truth of the statements in your argument easier.

But what if you want to declare more than one truth in your premise? In this case you can join two or more simple statements using a **statement operator**. A statement operator connects simple statements to form a **compound statement**. For example:

* Ann is home AND Bob is home
* The screen door has a hole in it OR I'm seeing things

In each sentence above, we have two simple statements joined by AND and OR and these operators allow us to declare more than one truth using simple statements. We'll learn a set of operators in a later lesson and you'll see why using operators is important when you start to construct your own arguments.

## Something is Missing

At this point, you may notice that something is missing. If you've ever tried to convince someone of something using an argument, you may notice that the argument may not work. The other person may not understand what you're saying or you may use the "wrong" word. You get bogged down in semantics rather than focusing on the logic of the argument.

Suppose I said,

"The epistemic position of the defense side of the litigation lacks justificatory veracity."

This is a declarative statement but it may not be fully understood by the person hearing the claim. I could simplify this statement as,

"The defendant failed to make her case."

## Introduction to Logic: Propositions

We learned in the last couple of lessons that we construct arguments using statements that declare something to be true. But what if we need to rephrase a statement or translate the statement into another language? Does this change the argument?

Let's look at an analogous problem in metaphysics in order to tackle this problem.

Sidebar: Philosophical metaphysics is the study of being and existence. It comes from the Greek words "meta" -- above and "phusike" -- matter: that which comes before the study of the physical world. The discipline attempts to come up with definitions for existence and studies things like relations, identity, change, cause and effect and similar subjects.

## The Number Two

What is this:

### 2

If you said "the number 2" you'd be wrong. That's actually the numeral 2. Notice that you can take a pen and paper and write a numeral 2 on that paper or you may have many numeral 2s on the calendar on your wall. What do all these numeral 2s have in common? They represent, symbolize or point to something else: the number 2!

Metaphysicians make this distinction by examining each thing's properties. Properties are just the aspects of things that make them the same or different. Suppose you took a red marker and drew a numeral 2 on a piece of paper about 5 inches tall. That numeral 2 has specific properties. It is:

* Red
* Five inches tall
* Made up of chemicals (the chemicals of the marker)
* It's even located somewhere: on your desk which can be determined by GPS!

Now ask yourself: does the number 2 have those properties? Is the number 2 red? Is it 5 inches tall? Is it located somewhere? The answers to all these questions seem to be no. So the numeral 2 on your desk seems to be different than the number 2. What are the properties of the number 2? Well, we can say it's even and greater than 1 and less than 3 and can be used in mathematical operations like addition. None of these properties seem to be true of the numeral 2.

So these property differences give us at least a superficial reason to think the numeral 2 is a symbol for the number 2 and that the two things are different. To emphasize this, notice that you can symbolize the number 2 in a variety of ways: using dots, your voice, or chickens. As long as the thing you're using points to the number, you can symbolize it almost any way you like!

Okay, we acknowledge that this story requires that you buy into a lot of complicated metaphysics that are hotly debated by philosophers and mathematicians alike. But for the purposes of this lesson, it's not important that you understand or agree with all of the metaphysics presented above. The main takeaway is that you can see the difference between a symbol and the thing the symbol symbolizes. This is going to relate directly the problem we introduced at the beginning of this lesson.

## Introduction to Logic: More on Propositions

The last lesson taught us that a variety of symbols can be used to point to something common. I can use stars, \*\*, to symbolize the number 2 for example. This applies directly to our problem of statements and how they're used in arguments.

## Propositions

We saw earlier that statements are linguistic. That is statements essentially are sentences that are spoken or written in some language. These are not the things that the logical relation relates. They only symbolize the thing we care about in logic: propositions.

Philosopher Peter van Inwagen defines a proposition as a "non-linguistic bearer of truth value". Put more simply, a proposition is the meaning behind a declarative statement and it is the thing that is true or false. A statement then symbolizes some proposition.

Why is this important? Consider the following:

1. Pat likes Jan
2. Jan is liked by Pat

These are two different English sentences. The first has three words and the second has 5. In the first sentence, the word "Pat" comes first and "Jan" is first in the second. You get the idea. The relevant question here is do they mean the same thing? Do the sentences symbolize or represent the same idea? The most straightforward answer is: yes they do. Here's another example:

1. The moon has craters
2. La luna tiene cráteres

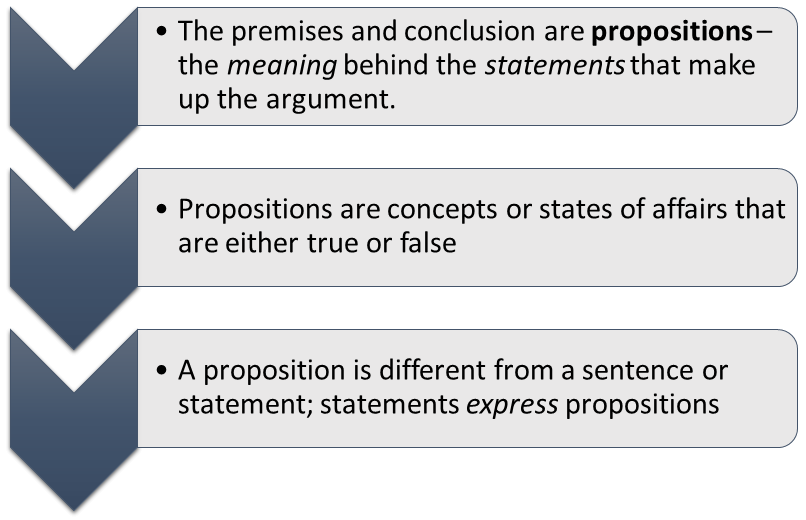
Here we have two sentences in different languages. But we would say the Spanish sentence means the same thing as the English sentence. Using the vocabulary of the last couple of lessons, we can say that both sentences symbolize the same thing, the same proposition.

## Why does this matter?

Let's return to the example from a previous lesson.

1. The epistemic position of the defense side of the litigation lacks justificatory veracity.
2. The defendant failed to make her case.

We can use these two statements in a logical argument interchangeably as long as they both express the same meaning. But for clarity, we might choose sentence 2 over sentence one because it's easier to understand (depending on the context and audience and other factors).



The value of understanding propositions is to prevent us from getting too bogged down in syntax (or word choice). While word choice can be very important, becoming overly attached to a word can sometimes derail getting at the broader meaning of what one intends to say.

One application of this principle is rewording an argument in order to make it easier to work with. We may need to reword a proposition to express the idea in simple sentences if the original argument wasn't written that way. As long as the rewording expresses the exact same idea--the same proposition-- this is allowed!

## Introduction to Logic: Truth Value

We've covered the structure of arguments and the make up of the individual components that make them up. Studying just the structure of an argument can get you pretty far but to evaluate an argument fully, you have to know whether its premises and conclusion are true. Logicians call this the propositions truth value.

In logic, a proposition can only be true or false. It can't be neither and it can't be both. When you don't know a proposition's truth value, you can, for the sake of analysis, assume it's true.

Many propositions can be true at one time and false at another and some are always true and always false. For example:

1. A square is a circle
2. Mount Fuji is located on Honshu Island in Japan
3. It is raining in Seattle

Proposition number 1 is always false and number 2 is always true. These are not dependent on any other condition and do not change. Proposition 3 can be true at one point in time and false at another (and during certain seasons, change from minute to minute!). So in order to clarify the proposition, we might have to specify the time. So we might need to rephrase proposition 3 as, "It rained in Seattle on May 5, 2017". Now we determine whether it did in fact rain in Seattle at that time and assign it the proper truth value.

But we can assign 3 a truth value as it is written even if we're not sure. We can assign it a value of true for the sake of argument and see how it works in an argument.

## Section Summary

In this section, we took a look at the basic structure of a computing device called a **Turing Machine**. We then briefly looked at how computers process information and related the basic functioning of a computer to formal logic.

Then we saw that **arguments** are the object of logical analysis and are made up of **premises**, a **conclusion** and a **logical relation**. The premises and conclusion of an argument are written as statements which are symbolic representations of meaning or what logicians call **propositions**.

Finally, we saw that propositions have a **truth value** which is the truth or falsity of a proposition at a given time. We now have the basic foundation from which we can begin to construct and analyze arguments. We'll dig into that further in the next module.

For the remainder of this module, we'll begin to explore one way we can apply formal logic to our lives and even to designing and developing computer programs: critical thinking.

## Symbolizing Deductive Arguments

In mathematics, we can use symbols in equations and formulas and we can replace those symbols with a variety of numbers and the equations still work. For example, we could create an addition formula using symbols:

x + 1 = y

We either could replace x with a number and solve for y or replace y with a number and solve for x.

Similarly, we can symbolize propositions in deductive arguments and from the symbolic form alone, we can better see the argument's structure.

## Symbolization at work

Suppose I want to argue that all humans belong to the class of things that are both rational and conscious. I might create this argument:

1. All humans are rational
2. All rational things are conscious
3. Therefore, all humans are conscious

Now let's do this:

Let H stand for human, R stand for rational, and C stand for conscious and write our argument as:

1. All H are R
2. All R are C
3. Therefore All H are C

Notice that we can also replace H, C, and R with any term and it doesn't change the overall structure of the argument. Replace H with "haberdashery", R with "ripe" and C with "crazy". We now have:

1. All haberdashery are ripe
2. All ripe things are crazy
3. Therefore all haberdashery are crazy

The argument is non-sensical but it retains the same structure and we'll see why that is important in an upcoming lesson.

So using symbols, we can separate the structure of an argument from its content and this will be key to breaking down arguments for analysis.

## Introduction to Operators

As we saw in a previous lesson, simple statements can be joined to other simple statements to create compound statements. A simple statement declares that something is true. In order to make arguments easier to manage and determine the truth value of each part of your argument, you should break down all statements to simple statements whenever possible.

For example, take this statement:

Modern computer systems are Turing machines that can process trillions of operations per second though the basic operations are extremely simple: they essentially transition states from 1 through 4 or on, semi-on, sort-of-on, and off.

How many truth statements are made in this sentence? If you were to give a single truth value to the entire statement, what would it be? If you are having difficulty figuring that out, try this rewording:

Modern computer systems are Turing machines AND modern computers can process trillions of operations per second BUT the basic operations of computer systems are extremely simple AND modern computer systems transition states from 1 through 4 or on, semi-on, sort-of-on, and off.

Now it should be easier to see all the truth statements that are made. The "AND" and "BUT" words separate the simple statements. There are 4 separate statements of truth (with the truth value of each in parentheses:

1. Modern computer systems are Turing machines (T)
2. Modern computers can process trillions of operations per second (T)
3. The basic operations of computer systems are extremely simple (T)
4. Modern computer systems transition states from 1 through 4: on, semi-on, sort-of-on, and off (F)

Statement 4 is false because a Turing machine just transitions states between 1 and 0.

AND and BUT help us separate the simple statements and help us combine simple statements into a compound statement but we're able to easier manage the truth value of each simple statement. This is the value of using logical operators to construct compound statements.

## Monadic and Dyadic operators

Operators can be monadic or dyadic A monadic operator operates on a single simple statement. Other operators will all be dyadic operators because they operate on two simple statements. Any time you use an operator with a simple statement, even a monadic operator, you create a compound statement.

## Compound Statements and Truth Function

All the compound statements we'll look at are called "truth-functional" compound statements. This means that the truth value of the entire statement is determined by the truth value of the individual simple statements that make it up and the function of the operator on those statements.

If you write computer programs, you can think of the simple statements as inputs and the operator as the function that accepts those inputs and does something with them. The output of the entire function is determined by the value of the inputs and the specific operation of the function. Consider this simple program written in pseudo code which will take a string value and change the value to all caps and then print it to the screen:

Function PrintCaps (string A)

{

A = AllCaps(A);

Print A

}

If we send in the value "pencil" to our function, it will print "PENCIL" to the screen. What's important here is that the output is determined both by the value we pass to the function and the operation that is done to that value by the function itself. In truth-functional logic, operators work on simple statements (the function) and determine an overall truth value for the compound statement based on the truth value of the simple statements (inputs) it operates on. This will become clearer as you learn each of the operators and how they are used to create truth-functional compound statements.

## The Negation Operator

The negation operator is the only monadic operator and it negates the truth value of the simple statement that follows it. So if the simple statement is true, putting a negation symbol in front of it makes the compound statement false. If the simple statement is false, putting a negation in front of it makes the compound statement true.

The negation operator performs the operation of a Turing Machine we saw earlier in the module. Here's another example where computer science and truth-functional logic overlap.

**Symbol**: ~

**Symbol Name**: tilde

**Parts of negation**: A simple statement with a tilde preceding it

**Example**: Let's suppose we have the simple statement "Ann is home" and we symbolize that simple statement with the variable "A". Using negation with that simple statement would be:

~A or **It is not the case that** Ann is home

The wording used here is very important. In common English, if we wanted to negate the statement "Ann is home" we would say "Ann is **not home**" But notice when we do that, we change the simple statement. In order to keep the simple statement consistent, we want to keep the operator distinct from the statement which it operates on. This is why we say, "it is not the case that" and then we state the simple statement. Now A always refers to "Ann is home" and the negation operator is denoted by "it is not the case that" or ~A.

## Conjunction

The conjunction operator creates a compound statement such that in order for the whole statement to be true, each simple statement has to be true. This should make intuitive sense as you see how conjunction works.

**Symbol**: & or · . You can use either symbol for conjunction (some logic text books use one and other textbooks use the other)

**Symbol Name**: Ampersand or dot

**Parts of Conjunction**: Two simple statements joined by the conjunction symbol. Each simple statement in a conjunction is called a **conjunct**.

**Example**: For the two simple statements "Ann is home" (A) and "Bob is home" (B) a conjunction would look like this:

A & B (A · B) or Ann is home AND Bob is home.

As we said above, if either conjunct is false, the entire conjunction is false. This should make intuitive sense. If Ann is home but Bob is not home we would say that it isn't true that both Ann and Bob are home--which is what the conjunction compound statement is claiming.

At this point let's revisit the idea of truth-functional compound statements we learned in an earlier lesson. We can see how the individual simple statements have a truth value but the compound statement as a whole also has a truth value based on the work the operator does on those simple statements: the simple statements are the "inputs" and the conjunction is the "function" that operates (in truth value terms) on the simple statements to produce a single truth value.

## Disjunction

The disjunction operator creates a compound statement that is true if either simple statement is true but false if both simple statements are false.

**Symbol**: v . This is the letter V on English keyboards

**Symbol Name**: Wedge

**Parts of Disjunction**: Two simple statements joined by the disjunction symbol. Each simple statement in a disjunction is called a **disjunct**.

**Example**: For the two simple statements "Ann is home" (A) and "Bob is home" (B) a disjunction would look like this:

A v B or Ann is home OR Bob is home.

With disjunction we can see that if either disjunct is true the compound statement is true because "or" communicates that either could be true. But if both disjuncts are false the disjunction is false.

Note: If you've done certain types of math or computer programming, you might be familiar with "exclusive or" or XOR which states that the entire statement is true only if each disjunct differs in truth value (one is true and one is false). Disjunction is not XOR. Disjunction is true if either disjunct is or both disjuncts are true and false only if both are false -- sometimes called "inclusive or".

## Conditional Operator

As the name implies, the conditional operator creates a compound statement that sets up a condition for something to be true. If the condition is met, the statement is true.

**Symbol**: ⊃ or →. On keyboards, you can use the greater than symbol for the horseshoe: > or two dashes and the greater than symbol for the arrow: -->

**Symbol Name**: Horseshoe or arrow

**Parts of Conditional**: Two simple statements joined by the conditional symbol. The first simple statement in a conditional is called the **antecedent** and the second simple statement is called the **consequent**.

**Example**: For the two simple statements "Ann is home" (A) and "Bob is home" (B) a conditional would look like this:

A ⊃ B (A → B) or IF Ann is home THEN Bob is home. Note that the simple statements do not include the IF and the THEN--those are the operator keywords.

## More on conditionals

Conditionals provide us with some interesting logical relationships. Remember that conditionals are not arguments in and of themselves but can make up premises (or parts of premises) of an argument. Conditionals do tell us though what must necessarily be true if something else is true: it tells us that if the antecedent is true, the consequent must be true.

## Necessary and Sufficient conditions

This property of conditionals define what logicians call **necessary and sufficient conditions**. The necessary condition is what must be true or what is necessary. The sufficient condition tells us what could be true or one possible condition that can bring about the consequent. In other words, the antecedent is sufficient to bring about the consequent but other conditions can as well. For example, consider the following conditional:

If Alan Turing died in 1954, then he was born in or prior to 1954.

The sufficient condition--the antecedent--is: Alan Turing died in 1954. If that's true, then it must be true, it's necessarily true, that Alan Turing was born in or prior to 1954--the consequent. This is why we generally use conditionals to define terms or state what is necessarily true of something. What about this conditional:

If S is a human, then S is . . .

How would you fill in the consequent? What is necessarily true for something to be human? Does it have to have brown hair? What about being female or over the age of 30? Those don't seem to be necessary conditions for being human. Think about what is necessary and when you come up with that list (which probably is going to be conjunction compound statement), you have your basic definition of "human."

## Conditionals and truth conditions

A conditional is false only if the antecedent is true but the consequent is false. Under all other truth conditions, we say the conditional is true. To understand why, let's take a look at all the possible truth conditions for a conditional statement.

| **Truth Conditions** | **Example** | **Conditional as a whole** |
| --- | --- | --- |
| antecedent is true; consequent is true | If a byte is 8 bits, then a byte is greater than 7 bits | True |
| antecedent true; consequent false | If byte is 8 bits, then a byte is greater than 9 bits | False |

Why the truth condition of the conditional is true in the first example and false in the second should be intuitively clear--this is how we understand how an if-then statement works in English grammar. But what about these:

| **Truth Conditions** | **Example** | **Conditional as a whole** |
| --- | --- | --- |
| antecedent is false; consequent is true | If byte is 7 bits, then a byte is greater than 7 bits | True |
| antecedent is false; consequent is false | If a byte is 7 bits, then a byte is greater than 10 bits | True |

In order to understand why the conditional as a whole is true in these cases, we have to go back to the idea of the truth function of compound statements we saw in an earlier lesson. Truth function tells that the compound statement as a whole is a function of the truth value of its "inputs" or the simple statements that make it up. The statement as a whole is false only if the function "fails" or doesn't "accomplish" the goal of the function of the operator. Since the conditional tells us that if the input of the antecedent is true then the input of the consequent also must be true, the conditional "function" only fails if this turns out not to be the case. In all other cases, the function has not failed because the fail condition has not been met so the conditional can be considered true.

#### Okay, But That Doesn't Really Make Sense

If that's what you're thinking, here's a metaphor to help explain how this works:

Suppose you design an airport security system to detect necessary conditions. You design the system so that when a person passes through the security system carrying a cupcake, the security system states a fact about Alan Turing. As a security expert, the only thing you care about is that when a cupcake is brought through the screening area, the system must state something true about Alan Turing. You care only about that case as a whole. You don't care about it detecting cupcakes only; you care about it stating true facts about Turing when a cupcake is brought through. In all other cases, you don't care--the system can state any fact it wishes, true or false. Put another way, the system fails to function properly only if someone brings a cupcake through the screen and the system makes a false statement about Alan Turing.

So the conditional would be: if cupcake, then true fact about Alan Turning. The "carrying a cupcake" is like a true antecedent and the "truth about Alan Turing" is like a true consequent.

**Case 1** (antecedent is false; consequent is true): So let's suppose a person walks through the screen carrying a cookie. As the person passes through, the system says, "Alan Turing was born in 1912". The system happened to state a true fact about Turing but it doesn't matter because a person carried a cookie through the system. Since the person was not carrying a cupcake, the system can say anything it wants about Turing so the system is working properly in this case.

**Case 2** (antecedent is false; consequent is false): A person walks through carrying a banana and the system states, "Alan Turing died by consuming Nightshade." In this scenario, the system stated something false about Turing but because the person was carrying a banana, it doesn't matter. The system was working as expected because on a false antecedent (anything other than a cupcake), the system can say anything it wants.

**Case 3** (antecedent is true; consequent is true): A person walks through the screen carrying a cupcake and the system states, "Alan Turing worked at the National Physical Laboratory". Here we have our condition met in the antecedent and the system stated a true fact about Turing. So the system is working.

**Case 4** (antecedent is true; consequent is false): A person walks through with a cupcake, and the system states, "Alan Turing, studied at Oxford University." We have the condition in the antecedent that we're looking for but the system made a false statement about Turing. So the system is not working as we want it to--it failed to function properly based on what we want it to do.

This illustrates the idea of a "truth function." The conditional has to be taken as a whole and fails only under one scenario. In all other scenarios, we say the conditional "works" or is true.

Here's a computer program (written in pseudo code) that illustrates what's going on. For the inputs to the function, let's suppose that we can only pass in a 1 or a 0 and a 1 represents true for the simple statement and 0 represents false.

Function WhatsTheTruthValue (a,b)

{

if (a > 0){

var t = 3\*b;

if (t === 0){

print "function crashed";

return;

}

}

print "function works";

}

In this program, we pass values (either 1s or 0s) into the function and the function attempts to multiply 3 by the value of b. If that operation produces a value of 0, the function prints "function crashed" to the screen and doesn't go any further. Otherwise it prints "function works" to the screen. Now let's look at all the possible operations of this function.

| **Inputs** | **Function outputs** |
| --- | --- |
| a = 1; b = 1 | "function works" |
| a = 1; b = 0 | "function crashed" |
| a = 0; b = 1 | "function works" |
| a = 0; b = 0 | "function works" |

Because our function only multiply 3 by the value of b when a is 1, that's the only condition when it can possibly crash. So it crashes when only one condition is true: a is 1 (true) and b is 0 (false) causing the division operator to produce a value of 0. The important part here is that the output is a product of the inputs, outputs and the operation of the function as a whole.

## Sidebar: Operator of the Largest Scope

Before we leave this topic, we have one more important concept to cover. As we've seen, it's possible to combine operators and simple statements together to build rather complex compound statements. When you combine operators, it may be difficult to tell what type of statement the statement as a whole is. For example, what kind of statement is this:

1. p --> q & r v s & t

There are four operators but it's not possible to tell which is the main operator. It's important to know which is the main operator because that's how we determine the truth value of the statement as a whole.

Now let's write it this way:

1. p --> (q & ((r v s) & t))

Some logicians don't like parentheses next to each other so prefer to use brackets and would write the statement 2 as:

1. p --> (q & [(r v s) & t])

Now can you tell what kind of a statement this is? If you said "conditional" you're right. Similar to mathematics, we solve for the truth value of the statements in the innermost parentheses first working out and then can solve for the statement as a whole. The last operator you solve for is the operator of the largest scope and determines what kind of compound statement you're working with. To see this, let's assign the following truth values to our variables:

p = true; q = false; r = false; s = true and t = true

If we want to determine the truth value of the statement as a whole we would start with the innermost disjunction since that's the innermost compound statement.

* If r is false or s is true then the disjunction as a whole is true.
* Now we can solve for the conjunction involving the 't' since that's at the second "layer". Since the disjunction is true and t is true, then the conjunction as a whole ((r v s) & t) is true.
* Now we can solve for the next conjunction involving the q. Since the first conjunction is true and q is false, the conjunction as a whole is false.
* Finally, we can solve for our conditional. Since the second conjunction (which makes up the consequent of the conditional) is false and p (which makes up the antecedent) is true, our conditional rule tells us that the conditional as a whole is false.

So the statement as a whole is false. This is why knowing the operator of the largest scope is important and using parentheses and brackets can help. By the way, you use the same device in computer programs to help determine which operations must be performed first.

## Truth Tables

We're not going to cover truth tables at any length in this course but we can briefly mention them in this context because they apply to helping with this problem. A truth table is, as the name implies, a table by which you can organize statements and operators to determine the truth value of the operator of the largest scope. When trying to solve for the operator of the largest scope, we "build" a truth table by starting with what you know and slowly filling in the blanks. For example, a truth table where we'd solve for the truth of the compound statement above would start out like this (the first columns "declare" the variables and their values and are there mainly for reference):

1. Fill in the variable values

| **p** | **q** | **r** | **s** | **t** |  | **p** | **-->** | **(q** | **&** | **[(r** | **v** | **s)** | **&** | **t])** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| T | F | F | T | T |  | T |  | F |  | F |  | T |  | T |

1. Fill in the value for the compound statement of the smallest scope: the disjunction.

| **p** | **q** | **r** | **s** | **t** |  | **p** | **-->** | **(q** | **&** | **[(r** | **v** | **s)** | **&** | **t])** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| T | F | F | T | T |  | T |  | F |  | F | T | T |  | T |

1. Fill in the truth value for the compound statement next in line: the conjunction involving the t. Notice we use the truth value of the disjunction we just solved for in order to solve for the value of the conjunction as a whole.

| **p** | **q** | **r** | **s** | **t** |  | **p** | **-->** | **(q** | **&** | **[(r** | **v** | **s)** | **&** | **t])** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| T | F | F | T | T |  | T |  | F |  |  | T |  | T | T |

1. And next

| **p** | **q** | **r** | **s** | **t** |  | **p** | **-->** | **(q** | **&** | **[(r** | **v** | **s)** | **&** | **t])** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| T | F | F | T | T |  | T |  | F | F |  |  |  | T |  |

1. And finally, the operator of the largest scope which is our conditional!

| **p** | **q** | **r** | **s** | **t** |  | **p** | **-->** | **(q** | **&** | **[(r** | **v** | **s)** | **&** | **t])** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| T | F | F | T | T |  | T | F |  | F |  |  |  |  |  |

In each stage I removed the values for the individual variables once we solved for the operator just to show you how to think through the process. In real truth tables, you'd leave them in place so you can work backward to see how each value was derived. Using truth tables can help you visualize the work you need to do to find the truth value of the operator of the largest scope.

## Types of Arguments

Logic, as we have stated, is the study of the standards of correct reasoning. Logicians divide all arguments into two broad categories: deductive arguments and inductive arguments. Every argument, if clearly stated, can be classified as either deductive or inductive.

A **deductive argument** is put forward with the aim of proving its conclusion with complete certainty, in such a way that its conclusion must be true if its premises all are true. The author of a deductive argument (if sincere) intends to show conclusively, with complete certainty, that the conclusion must be true. An **inductive argument**, on the other hand, is put forward with a quite different goal in mind: An inductive argument is intended to show that its conclusion is probably or likely true, though not certainly true, if its premises all are true. The author of an inductive argument (if sincere) only aims to establish the intended conclusion with a high degree of probability although not with complete certainty.

The following arguments are deductive because each is intended to show conclusively that the conclusion must be true if the premises all are true:

Argument 1:

1. All whales are mammals.
2. All mammals are animals.
3. Therefore, certainly all whales must be animals.

Argument 2:

1. Dinner will be tacos or dinner will be burritos.
2. It is not the case that dinner will be tacos.
3. Therefore, it is certain we’ll have burritos for dinner.

Argument 3:

1. If the roof leaks, then the ceiling leaks.
2. If the ceiling leaks, then the carpet gets wet.
3. So, surely if the roof leaks, then the carpet gets wet.

In contrast, the following arguments are inductive because each aims to show only that the conclusion is probably, though not certainly, true:

Argument 4:

1. In all of recorded history, it has never snowed in Seattle, Washington in June.
2. Therefore, it is highly probable, though not totally certain, that it won’t snow in Seattle next June.

Argument 5:

1. The Rolling Stones have sold out every concert they’ve given over the past fifty years.
2. Therefore, it is quite probable that their next concert will sell out.

Argument 6:

1. It has been raining for the past ten days.
2. All the weather reports predict more rain tomorrow.
3. Therefore, although it is not absolutely certain, it is reasonable to conclude that it will rain tomorrow.

## Deductive and Inductive Indicator Words

If you are presenting an argument, and you want to make it crystal clear that your argument is deductive, place words in the conclusion that indicate the presence of deductive reasoning. Words such as must, necessarily, certainly, for sure, definitely, and absolutely indicate that you aim to conclusively establish your conclusion, in such a way that the conclusion must be true if the premises all are true. Notice that each deductive argument just given contains a deductive indicator word in its conclusion.

If you want to make it clear that your argument is inductive, use words in the conclusion that indicate the presence of inductive reasoning. Words such as “probably,” “likely,” and “it is reasonable to conclude” suggest that you intend your argument to show that the conclusion is probably, but not certainly, true, if the premises all are true. Each of the inductive arguments just given contains an inductive indicator word in its conclusion.

## Valid and Invalid Deductive Arguments

## Argument form

We've already learned that for deductive arguments, if the premises are true, then the conclusion must be true. But true premises are not enough for a deductive argument to yield a true conclusion. Consider the following:

1. If the water is at 100 degrees centigrade, then it boils
2. Water is made up of hydrogen and oxygen molecules
3. Therefore the water is boiling

Notice in this argument, both premises 1 and 2 are true but it doesn't necessarily mean that the conclusion is true. This is because the premises do not stand in the right relationship to each other. Now consider this argument:

1. If the water is at 100 degrees centigrade, then it boils
2. The water is at 100 degrees centigrade
3. Therefore, the water boils

Here's another:

1. All humans are rational
2. All rational things are conscious
3. Therefore, all humans are conscious

In these arguments, the two true premises are related properly so that the conclusion must be true given the truth of those premises. This structure is an example of what logicians call proper **argument form**.

## Valid and Invalid Forms

Deductive arguments in an argument form where true premises result in a conclusion that must be true are called **valid** arguments. Deductive arguments that are not in a valid form are called **invalid** arguments.

**Test for Validity:** Logicians have come up with a set of valid forms that we will study later but an easy test for validity is to ask yourself the following question: "If I assume the premises are true, is it possible that the conclusion could be false." If you answer "yes" then the argument is invalid and valid if you answer "no". One way to determine whether the conclusion can be false is to find a counterexample: a real-world situation where the premises are true but the conclusion is still false. We can call this the "intuitive test" for validity.

The above test for validity is foolproof and even works if you don't know the meaning of the terms in the argument. Consider this argument:

1. All grue are fizznit
2. Jabby is a grue
3. Therefore, jabby is a fizznit

Jabby, grue, and fizznit are nonsense terms that have no meaning but you can still see that the argument is valid. This is because the rule states that if you assume the premises are true and then determine that the conclusion must be true, the argument is valid. This shows us something important: its not the truth of the premises that makes an argument valid but rather the form its in. We can say this another way:

Deductive arguments with actually false premises can still be valid

In fact you can determine validity even if the argument doesn't have any terms at all. We briefly looked at this when we studied symbolization and we'll see how this works more deeply in the next section.

## Invalid arguments

In order to illustrate the concept of validity further, lets look at arguments where the premises **do not** lead to a certainly true conclusion because the structure is poor or **invalid arguments**.

1. All humans are rational
2. Some rational things are conscious
3. Therefore all humans are conscious

This is an invalid argument. Its similar to the valid argument above but did you notice what changed? We replaced the term "all" in the second premise with "some" and that made the argument invalid. In logic, the term "some" means "at least one" so the second premise can be reworded as "at least one rational thing is conscious. It could be the case that the one rational thing is not a human being so it's possible that the conclusion is false. Therefore the argument is not valid. Here's another example using a different argument form:

1. If you program computers, then you’re familiar with JavaScript
2. You’re familiar with JavaScript
3. Therefore, you program computers

Notice that the first premise tells us that it must follow that you're familiar with JavaScript if you program computers. It doesn't tell us anything about whether you program computers given that you are familiar with JavaScript. Here's another invalid argument in the same form that may make this point clearer:

1. If Alan Turing died in 1954, then he was born in or prior to 1954
2. Alan Turing was born in or prior to 1954
3. Therefore, Alan Turing died in 1954

Premise 1 tells us that if Alan Turing died in 1954 he must have been born in that year or prior to it but doesn't indicate any relationship the year of his birth and when he must have died. So the conclusion is possibly false even though the premises are true making the argument invalid.

## Sound Deductive Arguments

In an earlier lesson, we learned about deductive validity. Validity is important but its only one part of constructing a good deductive argument. Being in the right form is essential but we also want true premises. When our arguments are in a valid form and have true premises, we say our argument is **sound**. Constructing sounds arguments should always be your goal.

Let's take this argument:

1. All crayons are colored
2. All colored things are pigmented things
3. Therefore, all crayons are pigmented things

This argument is a valid form (you can tell because it's not possible that the premises are true but the conclusion false) and the premises are true so we have a sound argument.

Some other things to remember about soundness:

* All invalid arguments are unsound
* It's possible for an argument to be valid but unsound but not invalid and sound
* You can't determine soundness from the structure alone or with a argument that contains only symbols (though you can determine validity). This is because soundness depends on truth value and you can only know whether a premise is true if the premise has propositional content or meaning.

## Two Deductive Syllogisms

Now that we know what a deductive argument is and what makes a deductive argument valid or invalid, we can now learn some valid forms. Each of these syllogism are deductive arguments that you can use to construct your own arguments. Each is valid and if you examine the structure of each argument and apply the rule we learned, you should be able to see why.

## Modus Ponens

The modus ponens argument is an argument where you assert something is true. You "put forth" the proposition. So in this argument, you are in "the mode of putting" forth a true premise. That's what "modus ponens" means--the mode of putting. Here's the form:

1. If p then q (p --> q)
2. p
3. Therefore q

In the second premise, you assert as true the antecedent of the conditional of the first premise. This allows you to conclude the consequent of the conditional in the first premise. This is because the conditional in the first premise tells us that if p is true, q must follow with logical necessity. Here's an example:

1. If Jerry grows a mustache then George will grow a mustache
2. Jerry grows a mustache
3. Therefore, George will grow a mustache

## Modus Tollens

The modus tollens form is similar to the modus ponens except in the second premise, we're going to negate the consequent of the conditional in the first premise and this will result in negating the antecedent of the conditional in the first premise. Here's the form:

1. If p then q (p --> q)
2. It is not the case that q (~q)
3. Therefore, it is not the case that p (~p)

So in the second premise, we're taking away the consequent or we're in the "mode of taking" or negating the consequent. That's what modus tollens means: the mode of taking. Here's an example:

1. If you are taking a MOOC course then you are busy studying
2. It is not the case that you're busy studying
3. Therefore, it is not the case that you're taking a MOOC course

Notice that the first premise, the conditional, states that if you're taking a MOOC it is necessarily true that you're busy studying. Put another way, it's not possible that you're not busy studying if you're taking a MOOC. So, in premise 2, if we say you're not busy studying, then it's impossible that you're taking a MOOC.

## Sidebar: Two Formal Fallacies

In the previous lesson we looked at two argument forms: the modus ponens and the modus tollens. These forms are always valid and regardless of the values you use for 'p' and 'q', the argument will still be valid (though not necessarily sound). There are two fallacies that are related to these valid forms and briefly looking at them may help you understand how the valid forms work.

These fallacies are called formal fallacies because they are errors with the **form** of the argument. Formal fallacies are contrasted with informal fallacies which have to do with the **content** of the argument.

## Formal Fallacy 1: Affirming the Consequent

This fallacy is associated with the modus ponens argument form. As you'll recall, the modus ponens has the following structure:

1. p --> q
2. p
3. Therefore: q

We affirm that the antecedent of the first premise is true and that allows us to conclude that the consequent of the first premise also is true. The fallacious version of this attempts to affirm the consequent is true and then conclude that the antecedent is true. It looks like this:

1. p --> q
2. q
3. Therefore p

This is fallacious reasoning. Let's look at the fallacious version of the argument we looked at for the modus ponens argument form to see why:

1. If Jerry grows a mustache then George will grow a mustache
2. George grows a mustache
3. Therefore, Jerry will grow a mustache

Notice that the first premise doesn't tell us anything about what must be true if George grows a mustache. It only tells us that George will grow one if Jerry grows one. That's it. Because of this, affirming the consequent in premise two does not allow us to affirm the antecedent in the conclusion. We've reasoned fallaciously.

## Formal Fallacy 2: Denying the Antecedent

This fallacy is associated with the modus tollens argument form. Now that you understand how fallacies work, let's briefly look at this fallacy and see if you can understand why it's a fallacious form. First, go back and review the modus tollens argument form in the previous lesson then examine this fallacious form:

1. p --> q
2. ~p
3. Therefore ~q

Notice that in this form, we're attempting to deny the antecedent rather than the consequent as we do in the valid form. Now, replace 'p' and 'q' with actual statements (you can use the argument we looked at in the modus tollens lesson). Can you understand why this form is fallacious? Post a question in the discussion form if you're not sure.

## Two More Deductive Syllogisms

So far we've looked at the modus ponens and modus tollens. Both use conditional statements in the first premise. Next we'll look at an argument form that uses conditional statements in both premises and the conclusion.

## Hypothetical Syllogism

In this argument, a logical connection is made between the consequent of one conditional and antecedent of another. It also introduces a third term so in this argument we'll use 'p', 'q' and 'r' for our variables. Here's the form:

1. If p then q (p --> q)
2. If q then r (q --> r)
3. Therefore, if p then r (p --> r)

This argument tells us that if the antecedent of premise 1 is true then the consequent also must be true. And if we use that consequent as an antecedent in another conditional then the consequent of that second conditional also must be true. This allows us to affirm that if the antecedent of the first premise is true, the consequent of the second premise must also be true. Here's an example:

1. If she is as light as a duck, then she is made of wood
2. If she is made of wood, then she will float in water
3. Therefore, if she is as light as a duck, then she will float in water.

It's the necessary relationship between the antecedent and consequent of conditional statements that makes this argument work.

## Disjunctive Syllogism

The disjunctive syllogism is the last argument we'll look at in this course but be aware that there are many other forms you can study. We chose these four because they give you a solid framework for constructing logical arguments and have strong tie-in to logical constructions you will make in computer science.

The disjunctive syllogism is very simple. It takes the form of asserting a disjunction: either one thing is true or the other is. The second premise then denies that one thing is true and concludes that the other must be true. Here's the form:

1. Either p or q (p v q)
2. It is not the case that p (~p)
3. Therefore q

Note that in the second premise you can deny either disjunct. So we could have said in premise 2, it is not the case that q (~q). Just make sure that whatever disjunct you deny in premise 2, you affirm the other disjunct in the conclusion. Here's an example:

1. Either the Beatles are playing tonight or the Rolling Stones are playing
2. It is not the case that the Beatles are playing
3. Therefore, the Rolling Stones are playing

Premise 1 of this argument states that either one or the other must be true. In real life, we tend to think of myriad possibilities for a given situation and its rare that we only one of two options must be true. But that's what the disjunctive syllogism states. This is sometimes referred to as "an exhaustive dilemma" which means the two options (di meaning two and lemma meaning premise or proposition) are exhaustive-- or they cover all possibilities.

## Deductive Arguments and Computer Programs

If you've written a computer program before, you have recognize some of the argument forms that we've been looking at. Since computer programs are instructions that tell the CPU and other components what to do, the "flow" of the program is very similar to the flow of a logical argument. To see how this works, let's look at a couple of programs in different languages and compare them to our deductive argument forms.

## Testing for Login

The code in the function below, written in the JavaScript programming language, tests to see if a user is logged in and then presents a message on the screen welcoming them to the program.

function ProgramWelcome()

{

//check the user.loggedIn value to see if its true or false

if (user.loggedIn === true){

//if it's true, show the message

alert ("Welcome to the Program!");

}

}

Even if you've never seen JavaScript before, you may be able to make out what is going on. The function statement is just a wrapper for the code that does the real work. You can see a conditional statement in there, can't you? The statement says, if a variable is true (think of a variable as a container that holds a value), then show an alert. We can write the same "program" in logic:

1. If the user is logged in, then show the alert
2. The user is logged in
3. Therefore, show the alert

The computer program implements that same logical flow! Now let's add another statement to our program.

function ProgramWelcome()

{

//check the user.loggedIn value to see if its true or false

if (user.loggedIn === true || user.isFreeUser === true){

//if it's true, show the message

alert ("Welcome to the Program!");

}

}

The double pipe symbol (||) means "or" in many programming languages. Can you tell what that additional statement added to our program? We can write it out in logical form this way:

1. If the user is logged in or the user is a free user, then show the alert
2. The user is a free user
3. Therefore, show the alert

The argument would be symbolized this way:

1. (p v q) --> r
2. ~p
3. q
4. Therefore, r

In the first premise we have a disjunction as the antecedent and deny one of the disjuncts in the second premise. This allows us to assert the other disjunct in premise 3 and combine that with the conditional in premise 1 gives us our conclusion. Remember too that in a disjunction, if either disjunct is true, the whole disjunction is true. It works the same in the JavaScript program.

## Test for a number

Now let's look at a program in another language: C# (pronounced C sharp).

public bool isNumber (string p)

{

double num;

if(double.TryParse(p, out num)){

return true;

}

return false;

}

This program is a little harder to read if you're not familiar with C#. It simply tries to convert the string value we pass in (p) to a numeric value. If it parses properly (can convert the string to a number), then we know the string value p actually is a number. If it doesn't we know it's not a number.

We could write the same program using logical form:

1. if p can be parsed as a number, then p is a number
2. p can be parsed as a number
3. therefore, p is a number

AND

1. if p is a number, then p can be parsed as a number
2. it is not the case that p can be parsed as a number
3. therefore, it is not the case that p is a number

Notice that our computer program accounts for both the modus ponens and the modus tollens: the modus ponens is the "true" condition and the modus tollens is the "false" condition. Logic at work in C#!

## Strength and Inductive Arguments

Since deductive arguments are all about the necessary relationships between the premises and conclusion, the form is very important and we examined the form in the concept of validity and invalidity. As we saw in an earlier lesson, inductive arguments are all about the content of the premises and involve probability so validity doesn't apply to inductive arguments. Instead, logicians focus on the strength of inductive arguments.

## Strong Inductive Arguments

In a strong inductive argument, if the premises are assumed true, it is likely that the conclusion is true (though the conclusion could be false). If the conclusion is likely to not be true based on the assumed truth of the premises, the inductive argument is weak.

It's important to note that strength is a factor of to what degree the premises support the conclusion on the assumption that the premises are true and this is the case even if you know one or more of the premises are false. For example, the following is a strong inductive argument even though the premise is false:

1. All meteorites found to this day have contained gold
2. Therefore, probably the next meteorite found will contain gold

You may be able to see that the premise does support the conclusion to a high degree of probability and that's all we care about when we're determining strength.

## Weak Inductive Arguments

While there are mathematical ways to determine probability, most of the time you'll have to use your intuition and experience to help you figure out whether an inductive argument is strong or weak. For example, the following is a weak inductive argument:

1. This barrel contains one hundred apples
2. Three apples selected at random were found to be ripe
3. Therefore, probably all one hundred apples are ripe

You did not have to use Bayesian mathematics to determine that. You can just kind of "see" that its weak. We also have an intuition on how to make it strong:

1. This barrel contains one hundred apples
2. Eighty apples selected at random were found to be ripe
3. Therefore, probably all one hundred apples are ripe

Of course these are clear cases and things get more difficult as you get closer to the center of the probability scale (would 50 inspected apples make the conclusion more or less probable?) but many times, you have a sense of where the argument lands on the scale.

## Cogency and Inductive Arguments

A strong inductive argument with true premises is considered cogent. Like with deductive arguments, determining cogency is a two-step process. First determine the strength of the argument. A weak inductive argument is always uncogent. So if you can determine the argument is weak, you don't need to go any further.

If you determine that an argument is strong, then determine whether the premises are true. If they are, then you have a cogent inductive argument.

Here are examples of cogent inductive arguments:

1. It has rained over 20 inches per year in Seattle for last 20 years.
2. It will rain in Seattle again this year
3. Therefore it probably will rain over 20 inches over the course of the year.

Here's a cogent inductive argument in the form of a narrative:

"Each snow crystal contains about a quintillion water molecules, and these molecules can be arranged in countless numbers of ways. Of course, no one can prove it, but it is quite likely that every snowflake is unique." –Scott Camazine, The Naturalist's Year

## Determining the Strength of Inductive Arguments

Because inductive arguments are about the content of what you're arguing, determining strength is more subjective than determining validity. Put another way, the strength of inductive arguments is not based on rules but on judgment.

Even so, this doesn't mean that there aren't guidelines we can follow to help us assess whether some claim is more probably true or more probably false. In this lesson, we'll look at some of these guidelines.

When attempting to determine the strength of an inductive argument, you will use background knowledge and experience along with your belief on how that background knowledge impacts the statement in question. While there are scientific tools to help with this (see for example [Bayes' Theorem](https://en.wikipedia.org/wiki/Bayes'_theorem" \o "" \t "/home/ezequiel/Documentos\\x/_blank)), most of the time, you'll need to make a judgement call. However, there are some general patterns you can discover that can help.

For example, there is a class of statements where less information actually increases the strength of an inductive argument--that is, makes it more probable. You can determine this without knowing anything about the original sample size.

In his excellent book Thinking Fast and Slow, Daniel Kahneman uses the following example to illustrate this:

Linda is 31 years old, single, outspoken, and very bright. She majored in philosophy. As a student, she was deeply concerned with issues of discrimination and social justice, and also participated in anti-nuclear demonstrations. Which is more probable?

Linda is a bank teller.

Linda is a bank teller and is active in the feminist movement.

According to Kahneman, a study showed that the majority of people choose option 2 based on the description of the woman. But the general case is indeed more probable because the more relevant specificity (what you specify has to relate to what the claim is about) you add to this description, the less likely it is that a person will fit that description.

Put another way, there are more bank tellers in the world than there are bank tellers who also are in the feminist movement so without knowing anything more about Linda, it's more likely she'll be a part of the larger group. See [this Wikipedia article](https://en.wikipedia.org/wiki/Conjunction_fallacy" \o "" \t "/home/ezequiel/Documentos\\x/_blank) for more details. Of course, this does assume some background knowledge but it doesn't require that you know specific numbers or percentages to determine which is more probable.

In other cases, going from a general to a specific increases the probability of an inductive argument. For example:

John is looking to buy a new home automation lock for his house. John has the following requirements: The lock has to work with MagicHouse, a new home automation hub, it has to match his home aesthetics, and should not require a lot of maintenance. His friend Beth owns a LokTite, an automated lock brand. Since LokTite works well for Beth, John believes it should work for him. Which makes John's belief more probable:

Beth uses a home automation system

Beth uses the MagicHouse hub in her home automation system

In this example, because we learn that Beth uses the same automation hub that John uses--one of John's requirements--John's belief that LokTite will work is more probable given the information in statement 2. The additional information increases the probability that his belief will be true. This is because the original scenario specified what had to be true for John's belief to be true. Because we declare that this condition was met, the additional detail increases the probability that John's belief will be true.

Notice too that some additional propositions wouldn't affect the probability at all. Suppose your were told:

1. Beth's middle name is Jessica

This would have no impact on the probability that John's belief is true because the additional information doesn't relate to John's belief in any way.

## Modal Modifiers

In general, using the modal terms "necessary" or "impossible" (or their variants) will decrease the strength of an inductive argument (where "necessary" means it must be true without exception and "impossible" means it can't be true without exception). This is because necessary and impossible generally reduce the condition under which a statement can be true. For example:

Marwa likes a lot of different kinds of ice cream. Marwa likes most of the "standard" flavors you can find at most place that sell ice cream. Given that information, are the following statements more or less probable:

Marwa likes vanilla ice cream

Marwa necessarily likes vanilla ice cream

Hopefully you can intuitively see that statement 2 is less probable than statement 1. Because Marwa likes "most" of the standard flavors, we can't guarantee that she likes vanilla. It's probable but not necessarily true. Statement 2 says it would be impossible that Marwa not like vanilla. That's too strong and the probability that its true given our scenario is lower than claiming simply that she likes the flavor.

There are other factors that affect the strength of weakness of an inductive argument as well but the key thing to ask is whether the additional information you're being given make it more or less likely (or does it leave the probability unaffected) that the claim is probably true.

## What is Categorical Logic?

The ancient philosopher Aristotle (384-322 BCE) established many of the foundations of Western thought that many of us take for granted. In fact, his philosophy is so commonplace that it is difficult for us to imagine what the world would be like without it. For example, Aristotle laid the foundation for what we now know as modern science. He laid down the rules for scientific classification that in one way or another is essentially the framework we use today.

One of his achievements was to give us the framework for classification that is essential in biology, chemistry, and many other disciplines. For example, there are many types of cats in the world. You may have a variety of one of these ancient creatures as a pet. But there are many wild varieties like lions, cougars, lynx, and tigers that bear some resemblance to your house cat but differ in important ways. Still, we lump them all under the genus Felis. Cats also are categorized under the class mammalia and the order carnivora. This class and order include animals other than cats and have common characteristics that cats share that provide the basis for their membership into these classifications. Even if you've never formally studied biology, you're most likely familiar with classifications like these and its hard to imagine thinking about the world of biology without them.

But there was a time when a system like this wasn't even thought of. It was Aristotle that first set down the methodology for making classifications such as these and it changed the way humans saw the world. One of the tools Aristotle developed to help establish his methodology was an early logical system that has come to be known as categorical logic.

## The Logic of Categories

When you think about it, a genus, species, class, and order are all ways of categorizing things. You've most likely used a catalog at some point. If you've done any online shopping for clothes, you may have browsed a list of shoes and may have filtered the list to brown, leather shoes without laces. The catalog provides logical groupings or categories (from which the word catalog is derived) of the things you shopped for to make it easier to find what you're looking for.

You can think of a category as boundary: some things are inside the boundary, and other things are outside the boundary. The things that are inside the boundary--cats or shoes for example--all share some things in common. The things that don't have those shared characteristics are outside the boundary (but most likely are in another category shared with other things). Very simply, categorical logic is the science of analyzing groupings of ideas and discovering relationships between ideas using those groupings.

Consider this argument put in terms of categorical logic:

Premise 1: All cats are mammals

Premise 2: Some cats have fur

Conclusion: Therefore some mammals have fur

The terms all and some are categorical terms. They provide a vocabulary for talking about classes of things and using that vocabulary, we can establish relationships between terms (and the ideas those terms represent). In the above argument, we establish two categorical truths and it enables us to conclude a third categorical truth based on the truth of those premises. You might not find the above argument very informative but that's not really the point. The argument shows that using the logic of categories, we can learn rules for how terms and the ideas and things those terms symbolize are related to one another. As we've been learning, this is the magic of logic.

## The Language of Categorical Logic: Some, All, and None

As we saw in a previous topic, categorical logic uses just three terms to establish all the interesting relationships between terms that we care about: some, all and none (or no).

Using these terms, there are four basic patterns or forms that a categorical sentence might follow:

1. State that all the members of one category belong to a second category; for example, “All cats belong to the category of mammals,” or simply, “All cats are mammals.”
2. State that none of one category belong to a second category; for example, “No cats belong to the category of fish,” or simply, “No cats are fish.”
3. State that some members of one category belong to a second category; for example, "Some cats belong to the category of pets," or simply, "Some cats are pets."
4. State that some members of one category do not belong to a second category; for example, "Some cats do not belong to the category of pets," or simply, "Some cats are not pets."

Our example of the first pattern is, "All cats are mammals." Aristotle broke this type of categorical sentence down into four parts, named as follows:

**The quantifier**: All

**The subject term**: cats

**The copula**: are

**The predicate term**: mammals

All four sentence types follow this basic pattern and understanding this pattern will be important as we work through how these statements are used in arguments.

## All and None

The terms "all" and "none," as their use in English grammar denotes, quantifies or tells us the number of the subject term that belong to the class denoted by the predicate term. Thus, "all cats are pets" means that every cat that exists is a member of the category or class "pet" and "no cats are fish" means that no cat that exists is a part of the category "fish."

## Some

"Some aardvarks are pets" asserts that some members of the subject category (aardvarks) are included within the predicate category (pets). But what exactly does the word some mean? This seemingly innocent little word turns out, on close inspection, to be ambiguous. A definition is needed to make things more precise. Two interpretations are possible:

Some can mean “one or more but not all.”

Some can mean “one or more,” that is, “at least one.”

Aristotle chose the second meaning, and it has become the standard in logic. Thus, in standard categorical logic, the word some shall mean "one or more," which means the same as "at least one."

"Some aardvarks are not pets" says that some (one or more) members of the subject class are not included in the predicate class; that is, one or more aardvarks do not belong to the class of pets.

## Quantity and Quality in Categorical Statements

Aristotle discovered that every categorical sentence has a quantity and a quality. The quantity is universal if the sentence makes a claim about every member of the subject term category, and the quantity is particular if the sentence makes a claim about some members of the category denoted by the subject term.

Thus, a sentence such as, "All snakes are reptiles" is a universal sentence because it makes a claim about every member of the subject category. But a sentence such as, "No birds are mammals" is also universal, for it, too, makes a claim about every member of its subject category, namely, that every bird is not a mammal. In contrast, statements such as "Some cats are pets" and "Some cats are not pets" are obviously particular because they talk about some of the subject category.

The quality is affirmative if the copula affirms membership in the predicate category, and the quality is negative if it denies membership in the predicate group. For example, "All aardvarks are mammals" is affirmative, because the copula tells us the members of the subject category are included within the predicate category. "Some cats are pets" is also affirmative, for it, too, says members of the subject group belong to the predicate group.

However, the sentence "Some cats are not pets" is negative, since the copula denies membership in the predicate category (“Some cats do not belong to the pet category”).

Thus:

1. Sentences of the form "All S are P" are both universal and affirmative. They are therefore **universal affirmative sentences**.
2. Sentences of the form "No S are P" are both universal and negative. These are **universal negative sentences**.
3. Sentences of the form "Some S are P" are both particular and affirmative. These are **particular affirmative sentences**.
4. Sentences of the form "Some S are not P" are both particular and negative. These are **particular negative sentences**.

## Standard Categorical Form

You've been learning in this course that part of the goal of logic is to take arguments from everyday grammar and shape it into logical form so it's easier to analyze. For example if someone said to you:

"You'll wait in traffic longer by going that way."

You can restate this as a compound statement, a conditional:

"If you go that way then you'll wait in traffic longer."

Each sentence means the same thing but by restating the sentence as a conditional, you can now place it into one of our standard forms and use it in an argument. Statements in categorical logic have a standard form as well and we'll look at that briefly.

## Standard Form

The subject and predicate terms of a standard categorical statement must each be a noun that denotes a class or category of things. Thus, the sentence “Some roses are red” is not in standard form, since its predicate term is an adjective (red), and adjectives do not by themselves name groups of things (they denote characteristics of things). In everyday life, we often state categorical sentences that have adjectives as their predicate terms, but this is easy to fix. If the predicate term is only an adjective, we can change the adjective to a noun or a noun-like expression referring to a class or category of things. For example, if we rewrite “Some roses are red” as “Some roses are red flowers,” then the sentence fits the standard form, for red flowers names a class of things. In general, then, if a statement has an adjectival predicate, replace this predicate with a term naming the class of all objects of which the adjective may truly be predicated. Study the following examples closely:

| **Nonstandard** |  | **Standard** |
| --- | --- | --- |
| All tigers are carnivorous. |  | All tigers are carnivorous animals. |
| All deer are fleeing the fire. |  | All deer are things that are fleeing the fire. |
| All students are striking. |  | All students are persons who are striking. |

### Quantifiers

Sometimes we have to restate a sentence to include a quantifier. This can be tricky because the quantifier may be implied but it's not always clear which one to use. To decide, think about what the author probably intends, and then add the quantifier the author probably intended. Take a look at these comparisons:

| **Nonstandard** |  | **Standard** |
| --- | --- | --- |
| A tiger is a mammal. |  | All tigers are mammals. |
| A fish is not a mammal. |  | No fish are mammals. |
| Emeralds are green. |  | All emeralds are green things. |
| A whale is a beautiful creature. |  | All whales are beautiful creatures. |

The "some" quantifier can be even trickier because there are many words people use to denote "at least one." As a rule, any quantity less than all is best translated as some:

| **Nonstandard Quantifier** |  | **Standard Interpretation** |
| --- | --- | --- |
| Most cats are cute. |  | Some cats are cute animals. |
| Few aardvarks are handsome. |  | Some aardvarks are handsome animals. |
| At least one platypus is cute. |  | Some platypuses are cute animals. |
| Several pigs are smart. |  | Some pigs are smart animals. |
| Many bears are timid. |  | Some bears are timid animals. |
| There are bears in the woods. |  | Some bears are animals in the woods. |
| A tiger roared. |  | Some tigers are animals that roared. |

### Copulas

The copula connects (couples) the subject and predicate terms. In standard form, the only copulas allowed are the words are and are not. Consider the sentence “All mice eat cheese.” This sentence obviously is meant to express a universal affirmative statement. However, the sentence does not contain are or are not. To place this sentence into standard form, we rewrite the predicate term so as to reserve the meaning of the original sentence while at the same time using the copula are: "All mice are cheese-eaters." In other words, all mice belong to the category of cheese-eating creatures. For another example, "All hens lay eggs" is not in standard form since it lacks a proper copula. Adding the copula are and making suitable adjustments produces: "All hens are egg-laying animals."

## The Categorical Syllogism

Now that we have background into categorical logic, we can start to construct categorical syllogisms. Remember from a previous lesson that syllogisms have exactly two premises and one conclusion. To this definition, we'll also add the following properties of categorical syllogisms:

* Categorical syllogisms are composed only of categorical statements
* Each sentence in the argument contains exactly two terms—no more, no less.
* The argument as a whole contains exactly three different terms, each appearing exactly twice in the argument.
* No term appears twice in the same sentence.

The following argument is an example:

All whales are swimmers.

All whales are mammals.

Therefore, some mammals are swimmers.

## Middle, Major, and Minor Terms and Standard Form

The term appearing in both premises (and thus not in the conclusion) is called the **middle term**. The predicate term of the conclusion is called the **major term**. The term appearing as the subject of the conclusion is the **minor term**. The **major premise** is the premise containing the major term, and the **minor premise** is the premise containing the minor term. In the following example, animals is the major term, snakes is the minor term, and pets is the middle term.

1. All pets are animals

2. Some snakes are pets

3. So, necessarily, some snakes are animals

**A categorical syllogism is in standard form if the major premise is listed first, followed by the minor premise, and finally the conclusion**.

Because categorical syllogisms follow these strict rules of form, there is a limited, valid set of arrangements they can be in and we'll look at those arrangements or forms in the next topic.

## Forms of Categorical Syllogisms

During the Middle Ages, logicians assigned Latin names to the fifteen forms of syllogism first proved valid by Aristotle. These syllogisms were arranged into four groups known as “figures.” The exact logical form for a syllogism is specified by giving the type (A, E, I, or O) for each sentence followed by the number of the syllogism’s figure.

It's not necessary that you memorize all these forms but do become familiar with their general structure. You can use this topic as a reference when you're constructing your own arguments.

## Syllogisms of Figure 1

| **Name** |  | **Logical Form** |
| --- | --- | --- |
| Barbara |  | AAA-1: All M are P; all S are M. So, all S are P. |
| Celarent |  | EAE-1: No M are P; all S are M. So, no S are P. |
| Darii |  | AII-1: All M are P; some S are M. So, some S are P. |
| Ferio |  | EIO-1: No M are P; some S are M. So, some S are not P. |

## Figure 2

| **Name** |  | **Logical Form** |
| --- | --- | --- |
| Cesare |  | EAE-2: No P are M; all S are M. So, no S are P. |
| Camestres |  | AEE-2: All P are M; no S are M. So, no S are P. |
| Festino |  | EIO-2: No P are M; some S are M. So, some S are not P. |
| Baroco |  | AOO-2: All P are M; some S are not M. So, some S are not P. |

## Figure 3

| **Name** |  | **Logical Form** |
| --- | --- | --- |
| Darapti |  | AAI-3: All M are P; all M are S. So, some S are P. |
| Disamis |  | IAI-3: Some M are P; all M are S. So, some S are P. |
| Datisi |  | AII-3: All M are P; some M are S. So, some S are P. |
| Felapton |  | EAO-3: No M are P; all M are S. So, some S are not P. |
| Bocardo |  | OAO-3: Some M are not P; all M are S. So, some S are not P. |
| Ferison |  | EIO-3: No M are P; some M are S. So, some S are not P. |

After Aristotle’s death, his successors at the Lyceum studied syllogisms in the overlooked fourth figure and showed that the following forms are valid.

## Figure 4

| **Name** |  | **Logical Form** |
| --- | --- | --- |
| Bramantip |  | AAI-4: All P are M; all M are S. So, some S are P. |
| Camenes |  | AEE-4: All P are M; no M are S. So, no S are P. |
| Dimaris |  | IAI-4: Some P are M; all M are S. So, some S are P. |
| Fesapo |  | EAO-4 No P are M; all M are S. So, some S are not P. |
| Fresison |  | EIO-4: No P are M; some M are S. So, some S are not P. |

## Representing Categorical Statements Diagrammatically

As we saw in earlier lessons, deductive arguments can be valid or invalid. A valid deductive argument is an argument that is written in a particular form and follows the rule: if the premises are considered true, the conclusion is necessarily true. If the premises are consider true but there is at least one case where the conclusion is false, then the argument is not in valid form.

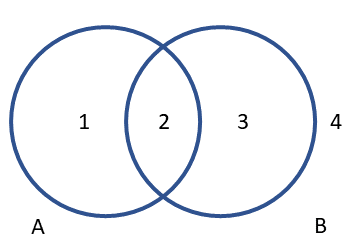
Categorical syllogisms adhere to the same principles. You learned in the last lesson that there are a set of valid logical forms for categorical statements and you can test a categorical syllogism for validity by verifying that the syllogism matches the structure of one of those forms. But there's another way to test for validity that doesn't require you to know or even reference those forms.

## The Venn Diagram

In the nineteenth century, the English logician and philosopher John Venn (1834– 1923) revolutionized categorical logic by inventing a radically different way to test categorical syllogisms for validity and invalidity. Venn’s method allows us to visually represent the information content of categorical sentences in such a way that we can actually see the relations between the sentences of a syllogism. This, in turn, allows us to determine validity and invalidity visually simply by looking at a diagram. When matters get extremely complex, it is always nice when we can draw a picture. This was an enormous leap forward for logical theory.

A Venn diagram is a set of overlapping circles, with each circle standing for a category or class of things, and with the circles arranged so as to display the logical relationships between the categories represented. A Venn diagram for a single categorical statement has two overlapping circles, one standing for the category named by the subject term and one representing the category named by the predicate term, producing four distinct areas, or regions.

Consider the following categorial statement, All As are Bs and the following diagram:



The A circle represents all the things in the universe that are A, while the B circle represents all the things that are B. If the As are aardvarks, the A circle represents all aardvarks; if the Bs are brown things, then the B circle represents all the things that are brown, and so forth. Thus:

Region 1 represents things that are A, but not B.

Region 2 represents things that are both A and B.

Region 3 represents things that are B, but not A.

Region 4 (the area outside the circles) represents everything that is neither A nor B.

## Creating Venn Diagrams for Categorical Statements

Information is entered into a Venn diagram in three ways: we shade an area to indicate that the area is empty. (Note that in many math contexts using Venn diagrams, shading means the opposite; i.e., that the quadrant is full.) We place an X in an area to say that the area contains at least one thing. In addition, we will adopt the following convention: If two areas have been defined and we know that something exists in one of the two areas but we do not know which of the two areas contains the item, we straddle the line, that is, we will place the X directly on the line separating the two areas. An X straddling a line will therefore indicate that (a) an item exists on one side or the other of the line but (b) the available information does not tell us which side of the line contains the entity in question.

Note: The Aristotelian Assumption The system of principles of categorical logic developed by Aristotle rested on a nonobvious assumption, namely, the assumption that the subject terms of categorical statements refer to actually existing things. This assumption is known in logic as the existential assumption. It is also called the “Aristotelian assumption,” the “traditional assumption,” and the “assumption of existential import.”

A statement is said to have existential import if its subject term refers to one or more actually existing things. A statement lacks existential import if its subject term does not refer to one or more actually existing things. For example, the statement “All unicorns are magical flying creatures” obviously lacks existential import because unicorns do not actually exist. We say that the subject term in this case (unicorns) is an empty term because the category it refers to does not contain any actually existing entities. In contrast, the statement “Some cats are pets” has existential import because at least one cat exists. The term cats is thus not an empty term. Aristotle thus assumed that all categorical statements under consideration have existential import—they are all about existing things.

In this course, our Venn diagrams will follow a convention that does not include the existential assumption for simplicity purposes. This follows the approach taken by logician George Boole (1815-1864) and is called the Boolean viewpoint.

In the next two topics, we'll look at diagrams for each of the four categorical statement forms.

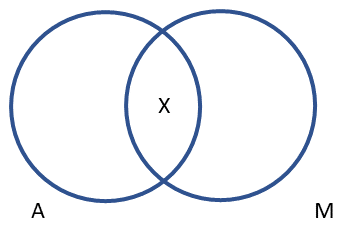
## Venn Diagrams for Forms I and O

In the previous topic we saw how Venn diagrams can represent the classes or categories of categorical statements. Since each of the forms have an established pattern, we can create Venn diagrams for each of the valid forms.

## Venn Diagrams for Particular Statements

### The 'I' Statement

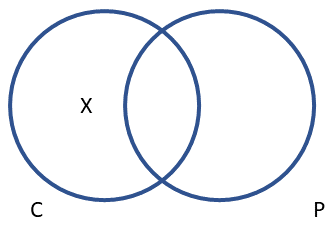
A particular statement is only true when its terms refer to one or more actually existing things, Using obvious abbreviations (A for “aardvarks,” etc.), the Venn diagram for the I statement “Some aardvarks are mammals” therefore looks like this:



You might find it helpful to imagine all the aardvarks in the universe rounded up and rooting around for ants inside the first circle—the A circle. Likewise, imagine all the mammals in the universe collected and resting inside the second circle—the M circle. The X indicates that there is at least one thing existing in the part of the aardvark region that overlaps with the mammal region. That is, some aardvarks are mammals. Notice that we did not put an X in region 3. Independent of the diagram, we might know that there are creatures in region 3—there are mammals that are not aardvarks—but the I statement does not tell us this, and we wish to diagram only the information the statement gives, and nothing more. Region 3 therefore gets no X.

### The 'O' Statement

The following is the Venn diagram for the O sentence “Some cats are not pets”:



Region 1 represents things that are cats but not pets, region 2 represents cats that are pets, and region 3 represents pets that are not cats. The X tells us there is at least one thing “inside” area 1.

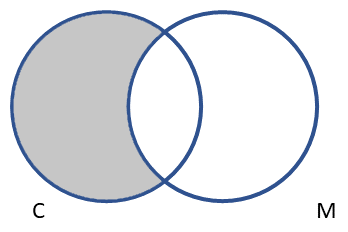
Note: If no X or shading appears in an area, this does not say that nothing exists in the area; rather, it indicates that nothing is known of the area. It only means we have no information about the area. Thus, for all we know, the area might be empty, or it might contain one or more things.

## Venn Diagrams for Forms A and E

## Venn Diagrams for Universal Statements

### The 'A' Statement

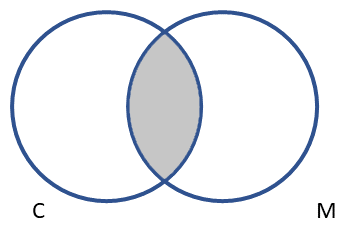
Venn diagram for the A statement “All cats are mammals,” looks like this:



Here we shade region 1 to represent an empty set. Namely there are no cats that are not mammals -- that category is "empty" because our A statement tells us that everything that is a cat is also a mammal.

### The 'E' Statement

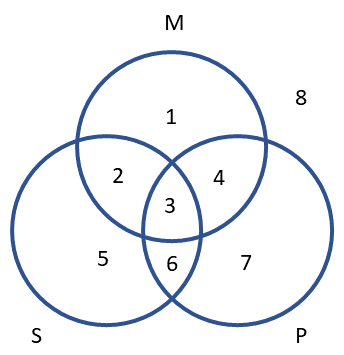
The corresponding universal negative or E statement in this case is understood as asserting, “Nothing is a cat and a mammal,” or “No cats are mammals.” The Venn diagram looks like this:



In this diagram, we shade only region 2 or the area that overlaps both cats and mammals. We do not shade region 3 because we're not making any claims about the entire class of mammals. Our E statement only tells us that if something is a cat it is not a mammal. So that category is null or empty.

## Using Venn Diagrams with Categorical Syllogisms

Recall that a categorical syllogism has two premises and a conclusion. A set of three interlocking Venn circles can be drawn to represent the information content and logical structure of an entire categorical syllogism. Once the diagram is drawn, we can determine, just from a visual inspection of the circles, whether or not the argument is valid. The first step is to set up the circles correctly. In the following example, S stands for the minor term, P stands for the major term, and M stands for the middle term. The circle labeled S represents the category of things referred to by the minor term, circle P represents the category of things referred to by the major term, and circle M represents the category of things referred to by the middle term.



To represent all the logical possibilities, the Venn circles must overlap to form the seven different areas numbered by convention shown here. Each of these areas, starting in order from the top, and dropping down and moving from left to right as one would read an English text, represents a different class of things:

Area 1: Anything here is an M, but not an S, and not a P.

Area 2: Anything here is an S and an M, but not a P.

Area 3: Anything here is an S, a P, and an M.

Area 4: Anything here is a P and an M, but not an S.

Area 5: Anything here is an S, but not an M and not a P.

Area 6: Anything here is an S and a P, but not an M.

Area 7: Anything here is a P, but not an S, and not an M.

Area 8: Things out here would be neither P, S, nor M.

In the next topic, we'll start to look at how to use Venn Diagrams to test categorical syllogisms for validity.

## The Venn Diagram Test for Categorical Syllogisms

In this and the subsequent few topics we'll walk through the steps for using a Venn diagram to test categorical syllogisms for validity.

Once your circles are in place, the following general procedure allows you to test a categorical syllogism for validity. We'll lay out steps 1-5 in this topic and then work through examples in subsequent topics.

## Creating a Venn Diagram for Categorical Syllogisms

1. Abbreviate the argument, replacing (consistently) each term with a single capital letter and retaining the quantifier and copula of each statement. (Example: “Some gems are not green rubies” becomes “Some G are not R.”) By convention we place the conclusion last.
2. Draw three overlapping circles, one for each term, to form seven distinct regions.
3. Label the circles using the three capital letters chosen (use the predicate term of the conclusion for the lower right circle, the subject term of the conclusion for the lower left circle, and the middle term for the middle circle).
4. Enter the information for both premises and stop. Enter only the information for the premises; do not enter information for the conclusion. If the argument contains only one universal premise, enter its information first. If the argument contains two universal premises or two particular premises, either premise can be entered first.
5. Finally, use the following tests to determine if the argument is valid or invalid.

A categorical syllogism is **valid** if, when the information from the two premises has been entered into the diagram, visual inspection of the diagram reveals that the information content of the conclusion is represented as well. In other words, by diagramming only the premises, we have also represented the information found in the conclusion.

Note: This shows that the information contained in the conclusion is already present in the premises. In a sense, the premises contain all the information presented in the conclusion. This in turn means that it would be impossible for the premises to be true and the conclusion false—a sure sign that an argument is valid.

A categorical syllogism is **invalid** if, when we have diagrammed the information content of the premises, information must be added to the diagram to represent the information content of the conclusion.

Note: If the diagram for the premises does not contain all the information in the conclusion, it is possible for the premises to be true and the conclusion false; that is, the conclusion could be false even though the premises are true. In this case, the argument is invalid.

## Creating a Venn Diagram: Valid Argument

In this topic, we'll walk through the process of creating a Venn Diagram for a valid argument to show how it works. Let’s begin with the Barbara (AAA-Figure 1) syllogism:

All mammals are warm-blooded creatures.

All aardvarks are mammals.

Therefore, all aardvarks are warm-blooded creatures.

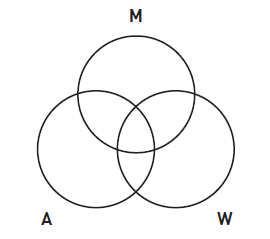
Our first step is to abbreviate the argument, placing the conclusion at the bottom:

All M are W.

All A are M.

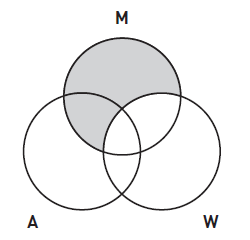
All A are W.

Second, we draw the three overlapping circles. For step 3 we label the circles:

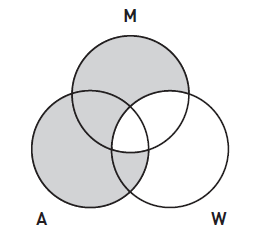


The subject term of the conclusion, A, is at the lower left. The predicate term of the conclusion, W, is at the lower right. The middle term, M, is above and to the middle.

Because both premises are universal, we can enter them into the diagram in either order. We’ll adopt the Aristotelian assumption in this example. The major premise tells us that all M are W. That is, no M are not W. In terms of the diagram, this indicates that all of the M circle that is outside the W circle is empty. Therefore, the part of the M circle that is outside the W circle must be shaded to indicate that it is empty.



Next, we enter into the diagram the information from the other premise, “All A are M.” This premise asserts that all aardvarks belong to the class of mammals. That is, all of the members of the A circle are inside the M circle. To enter this information, we must shade the part of the A circle that lies outside the M circle, indicating that this area is empty.



The preceding diagram represents the information contained in the premises. We purposely do not enter the information from the conclusion. Finally, we inspect the diagram to see if the information in the conclusion is already represented in the diagram of the premises. If it is, then the conclusion cannot possibly be false if the premises are true, which means the argument is valid. The conclusion “All A are W” indicates that all members of the class of A are also members of the class of W. That is, nothing in the A circle is not also in the W circle. And that is, indeed, exactly what our diagram indicates, for all sections of the A circle outside the W circle have been shaded. Nothing more would need to be added to our diagram to have it also represent the conclusion. The information content of the conclusion is already present in the premises. The conclusion would have to be true if the premises are true. **The argument is therefore valid**.

## Introduction to Critical Thinking

We're going to take a break from exploring formal logic to investigate how logic can be applied to analyzing and solving real-world problems. Critical thinking is a life skill that can (and we'll argue should) become a regular part of your way of thinking about the world. Like many skills, you probably already practice critical thinking without already knowing it and like most skills, with practice you can get better at thinking critically.

## What is Critical Thinking?

Although the word critical can be used to describe someone who is "inclined to criticize severely and unfavorably" (Merriam-Webster’s Collegiate Dictionary), the word has other uses. In academic contexts, critical thinking carries its original Greek meaning: "thinking based on criteria." A criterion (from the Greek word kriterion) is "a test," a "means of judging" (Dictionary.com), a "standard on which a judgment or decision may be based" (Merriam-Webster). Critical, or criterial, thinking is thus thought that is based on standards and vetted by tests.

Merriam-Webster gives another meaning that dovetails with the original Greek and that also forms part of what we usually have in mind when we speak of a critical thinker today: the word critical can also stand for "exercising or involving careful judgment or judicious evaluation" in “an effort to see a thing clearly and truly in order to judge it fairly.” This is also part of what we have in mind when we speak of a "critical care" nurse or a film "critic." Critical thought is therefore not simply negative or overly skeptical thinking, nor is it focused solely on finding fault or error. It is thinking guided by rational criteria that have proved to be reliable guides to good judgment and judicious evaluation.

## Why Study Critical Thinking?

The life you lead reflects, to a significant degree, your fundamental beliefs and values. The more critical your thinking, the more likely your fundamental beliefs and values correspond to reality, and consequently the more likely your life is rooted in reality rather than in illusion and falsehood. The study of the principles of critical thinking can sharpen your understanding of the criteria that have been found to lead to better judgments and that can help you see things as clearly and truly as possible.

Yet critical thinking is more than just following reliable criteria or proven standards presented in a textbook. It is as much an art as it is a science, as much a way of life as it is an exact form of thought. Socrates (470–399 BCE) was the first person in the historical record to devote his life to critical thinking by teaching it and by practicing it in a systematic way. He was also the first person to be put to death by a state for the "crime" of thinking critically. For these reasons, he is honored as the founder of critical thinking as a systematic discipline. It is appropriate that we begin our study with the life and thought of Socrates.

## Socrates the Critical Thinker

Socrates (470–399 BCE) stands out as one of the most influential figures in all of human history. Although he never wrote a book, led a conquering army, painted a masterpiece, or held high office, few individuals have made such a lasting impression on their fellow human beings and on the history of the world. By day, he supported his family, working (off and on) as a stonecutter, in the Greek city-state of Athens, the place of his birth. But his real profession, and his life’s passion, was the discipline the ancient Greeks had only recently named philosophy (from the Greek philein, "to love," and sophia for "wisdom," literally "the love of wisdom").

When Socrates was around forty, an incident occurred that would change his life forever. In ancient Greece, an oracle was someone believed to be channeling the gods. The most famous oracle in Socrates’s day was the priestess at the Temple of Apollo, located at Delphi.

People from all over Greece traveled to this sacred spot to ask the Pythia (her title) important questions. (The oracle in the movie The Matrix was based on the Oracle of Delphi. Alexander the Great traveled to Delphi to question the Pythia before embarking on his conquest of the Middle East.) When the session began, supernatural vapors would rise from the underworld, the Pythia (seated on a tripod and attended by temple priests) would fall into a trance, and the god would speak through her. (Modern scientists suspect the temple was located over a geological vent that released intoxicating underground gases that produced the effect. Archeologists have discovered chasms in the area releasing ethylene -- a gas that has been used as an anesthetic and that is actually capable of producing a trancelike state.)

One day, Chaerephon, a childhood friend of Socrates’s, traveled to Delphi to ask the oracle the following question: Is anyone wiser than Socrates? Her reply would become famous. “Of all men living Socrates is most wise.” Socrates’s first response upon hearing the prophecy was disbelief. Many years later, in a public speech given only days before the end of his life, he looked back on the event:

What could the god mean? . . . For some time I was frankly puzzled . . . but at last I embarked on my quest. I went to a man with a high reputation for wisdom -- I would rather not mention his name; he was one of the politicians -- and after some talk together it began to dawn on me that, wise as everyone thought him and wise as he thought himself, he was not really wise at all. I tried to point this out to him, but then he turned nasty, and so did others who were listening; so I went away, but with this reflection that anyhow I was wiser than this man; for, though in all probability neither of us knows anything, he thought he did when he did not, whereas I neither knew anything nor imagined I did. (Plato’s *Gorgias*, 495)

Socrates now began questioning people from all walks of life, from all around the city, about what they believed and why they believed it. He discovered that many people thought they knew all about some subject, when in reality they knew little about it. In fact, many were extremely ignorant. He also questioned experts famous for their knowledge of some subject matter, only to discover that although everyone believed these experts knew it all,and although they smugly believed they knew, they did not know what they were talking about. Their alleged expertise was a mirage.

Socrates believed that Chaerphon’s trip to Delphi and the message from the oracle were not accidents. He said, in utter seriousness, "The god has commanded me that I should live philosophizing, examining myself and others." With his eyes now opened to the pervasiveness of human ignorance and to the way it diminishes the human condition, Socrates had a new mission in life: to help people discover their own ignorance as a first step to the attainment of wisdom. To know nothing about a subject and to be aware of that ignorance, he argued, is better than to think one knows it all while actually knowing nothing. Knowledge of one’s ignorance is thus itself a form of wisdom.

Indeed, this knowledge, he now believed, is the first step to true wisdom, and to human excellence. Imagine what society would be like if each person pursued wisdom and excellence with fervor.

There are several important points to notice here.

1. Some people, upon hearing that they have been called the wisest person of all—and by a god no less—would become puffed up with pride. Not Socrates. He initially does not believe he is wise, and is not afraid to say so. This is the response of a man who is humble. Incidentally, Socrates does not intend us to take him literally when he says that he knows “absolutely nothing.” His words are best interpreted as an expression of his characteristic modesty.
2. Socrates does not take the Pythia’s statement for granted. He questions it, and most importantly, tests it (by going out into the marketplace and questioning people known for their wisdom). Socrates believes that questionable claims need to be tested by reasoning and observation, no matter who makes them, even if they come from a god.
3. Socrates makes his discovery about the pervasiveness of ignorance through conversations with others. Time and again he demonstrates that we learn a great deal about life, and about ourselves, in conversation with fellow human beings.
4. When Socrates investigates the matter, he does not go to an authority and beg hat in hand for an infallible answer. He looks, observes, reasons, and talks with people; in short, he searches on his own and thinks for himself.

Just as a builder must clear away brush before building a house, Socrates would say, one must clear away ignorance before building wisdom. Socrates was now convinced that self-knowledge is the basis for true wisdom and human excellence. In the marketplace, his conversations shifted from the big questions of cosmology and physics to the human condition, and to that which he now believed to be the most important question of all: What is the best life for a human being to live, all things considered? The Socratic method was taking form.

## Socrates and Truth

Reflecting on some of the lessons he had learned from the pronouncement of the oracle, Socrates later said:

And isn’t it a bad thing to be deceived about the truth, and a good thing to know what the truth is? For I assume that by knowing the truth [we] mean knowing things as they really are.

Two things are worth noting here. First, Socrates is claiming that knowledge of the truth is intrinsically good and ignorance is intrinsically bad. Philosophers distinguish between intrinsic goodness and extrinsic goodness. Something that is extrinsically good is good only insofar as it can be used as a means of attaining something else that is good. For example, unless one is a numismatist, a dollar bill has only extrinsic value. On the other hand, something is intrinsically good if it is good unto itself, if it is good completely on its own and not merely as a tool or means to get something else that is good. Socrates is claiming here that knowledge is good all by itself, regardless of what it may serve as a means to, and he is saying that ignorance is bad in itself, regardless of what it is used for or leads to.

Second, Socrates is defining truth. Truth, he is saying, is correspondence with reality, that is, with things as they really are. By implication, something is false, or illusory, if it does not correspond to reality. In philosophy, this account of truth, first stated explicitly by Socrates in Plato’s dialogues, is called the "correspondence theory of truth." More fully, according to the correspondence theory, a statement or proposition is true if it corresponds to reality; it is false otherwise. Put another way, a statement is true if it accurately describes or specifies reality; it is false if it does not. (As we saw in an earlier lesson, a statement or proposition may be defined as "that which is expressed by a declarative sentence," where a declarative sentence is "a sentence that says something that is either true or false.")

Putting these two points together, Socrates is claiming that it is intrinsically good to be in touch with reality, to be aware of things as they really are, rather than to be ignorant of reality. Truth is inherently better than illusion. He also says that this is an assumption he is making. Although it would be hard to prove this assumption true to someone who steadfastly denies its truth, it would be hard to live as if it were not the case that truth is superior to illusion, wouldn’t it? The notion of truth as correspondence with reality is part of common sense.

## The Socratic Method

One of the first things you notice when reading Plato’s dialogues is that Socrates asks his interlocutors lots of questions. However, his were not your usual questions. Socrates is out to help others discover their own ignorance as the first step to becoming the best persons they can be. To this end, he asks questions that will cause others to take a look in the mirror and critically evaluate their own assumptions, beliefs, values, and actions, and on the basis of independent, realistic criteria that have been found to be reliable guides to truth. Whether the individual is rich or poor,famous or unknown, of high or low class, the goal of Socratic questioning is always the same: that through self- examination, the individual will discover for himself or herself which of his or her own beliefs are true and which are not, which of his or her own actions are truly moral and which are unjust, and whether or not his or her own life is the best it can be.

Another thing that stands out, when reading Plato’s dialogues, is that his Socrates always engages people in conversation one-on-one. With a single exception (when he addresses a jury of his peers at the end of his life), his focus is always on the individual, never on the crowd.

This emphasis on self-examination explains why Socrates always expected honesty on the part of his interlocutors. If the other person is not answering honestly, then he or she won’t be led to examine his own beliefs and values. "Say what you really believe," he would sometimes tell those he was questioning. "Don’t play games with me!" For Socrates, honest self-examination was one of life’s most important tasks.

Sincere responses were also a requirement because if the interlocutor’s answers were not sincere, then the aim of the discussion would not be the unvarnished truth. At one point in a conversation recorded in Plato’s Gorgias, Socrates says to his friend, "Callicles, you’ll ruin our previous arguments and will no longer be examining the truth with me if you speak contrary to what you believe." (Plato’s Gorgias, 495) Moments later,Socrates has to remind him:

By the god of friendship, Callicles, you mustn’t think that you may play with me and say whatever comes into your head, contrary to your real opinion, nor, conversely, must you think of me as jesting. For you see what our discussions are all about—and is there [nothing] more serious than this: what is the way we ought to *live*? (Plato's *Gorgias*, 500 b-c)

Socrates questioned people especially hard when he thought they were in the grips of an illusion and needed a wake-up call. Why did he bother? The answer is that he believed in the value, dignity, and freedom of the individual. But this respect for the dignity of the individual also meant that if people who are deluded are going to wake up and become real, they will ultimately have to do it themselves—it cannot be done to them or for them.

At the same time, Socrates also saw that many people will never even start the process of honest self-examination by themselves: they need outside intervention. Just as an addict usually cannot quit all by himself, most people cannot shed their illusions all by themselves, either because they are unaware of their own ignorance, or because they don’t care, or because they are so emotionally invested in their illusions that giving them up is too painful.

So, Socrates concluded, people mired in illusion and self-deception need a nudge. In particular, they need someone who will question them until they look in the mirror and come to see for themselves their own ignorance and shortcomings. Critical thinking, as Socrates practiced it, was as much a form of therapy as it was a search for truth. This is ultimately why Socrates carried on his philosophical mission by means of one-on-one conversation.

**Socratic Suggestion**: On important issues, ask yourself the questions Socratic would ask: Are my assumptions on this matter really true? Or am I deluded? What does the evidence say? Are my actions morally right? Or am I only fooling myself? When others make questionable claims or propose questionable courses of action, ask them the same Socratic questions.

## The Socratic Method: Two Central Socratic Questions

Time and again, when he is in the midst of serious discussion with someone who is advocating a particular belief, or claiming that a particular action is morally right, Socrates asks two questions:

1. What exactly do you mean by that?
2. What evidence do you have for your claim?

In the various dialogues, we see Socrates asking these questions so often that they seem to be his two favorite questions.

**Question 1: What Exactly Do You Mean by That?**

Socrates would ask this question when he suspected that someone’s thinking was muddled. When our thinking is unclear, how can we know whether it does, or does not, correspond to reality? In many cases, after trying to answer Socrates’s question, the other person would discover that he really had no clear idea what he was actually claiming. In other words, the person did not really know what he was talking about.

When we reason with other people, we need to define our key words, or else they may not understand what we are saying. A **definition** is an explanation of the meaning of a word or phrase, and a precise definition is an exact explanation. Reasoning with words whose meanings are unclear can be like trying to play catch with puffs of smoke—hard to get a firm hold on anything. For these reasons, good definitions are an important part of the Socratic method, and of critical thinking generally.

**Question 2: What Is Your Evidence?**

The evidence for a claim is the reason (or reasons) for believing the claim is true, that is, for believing it corresponds to reality. People claim all sorts of things. Some people claim that aliens from a distant galaxy have secretly taken over the U.S. government. Others believe the Earth is actually a hollow sphere (with openings at the poles) and that an advanced civilization secretly inhabits the center of the planet. Interesting claims. However, before accepting controversial claims such as these, Socrates, if he were present, would ask for evidence.

Many times, after Socrates asked someone his second question, the person would discover that he actually had no solid evidence for the claim he was making; in modern terms, “He didn’t have a leg to stand on.” This is one of the reasons Socrates concluded that many people are deceiving themselves much of the time.

## Applying the Socratic Method to Computer Science

As we saw in previous lessons, critical thinking for Socrates is a life skill--something we can apply every day in a variety of situations. But we can apply critical thinking skills specifically to problems in computer science and to the business of computer science. Earlier we looked at how logic and computer design fundamentals work in similar ways. In this lesson, we're going to explore how critical thinking skills can be applied to writing and testing computer programs.

## Software Design

Software design is a large field and involves many disciplines. Microsoft has a course on software design which you can check out if you're interested [here](http://www.edx.org/school/microsoft" \o "" \t "/home/ezequiel/Documentos\\x/_blank). While each discipline involved in software design has a specific and different role to play, each person involved should apply critical thinking skills to ensure consistency and that all scenarios are covered.

Since the Socratic method is about asking enough of the right questions about the topic at hand, here are some questions that you can ask when designing software from scratch or making updates to existing software. As you read through these, think about how they are an example of the Socratic method and how you can apply the Two Socratic Questions to get further clarity on the answers.

| **Question** | **Description** |
| --- | --- |
| Who is the software for? | This question defines the audience you're building for and helps guide the design |
| Who is outside of our audience? | Engineers call this a "non-goal" and it defines who the software is not meant to reach |
| Why are we building this software | This may seem like an odd design question but it can force the team to think about the unique skills, abilities, and resources they bring to the table that will differentiate the software from others |
| What should the software do/not do? | Similar to the questions about audience, these are high-level goal and non-goal questions that set boundaries for what you want the software to accomplish. |
| Does a program already exist to do this? | This is a great critical question because it calls the whole project into question. |
| What is the simplest design that accomplishes the goals? | This question specifically focuses on design goals and not adding more to the design than is absolutely necessary |

These are just examples of the types of questions that software designers and engineers need to ask before and during the software development process.

## Software Testing

While the role "Software Tester" is less popular as it used to be (software engineers are now expected to do peer testing and use more automation), the attitude of the software tester should be one held by all software engineers.

When you put your testing hat on, you adopt a strongly critical perspective on the software you're building: you must assume there are flaws (known as "bugs") in the software and your job is to find them. This may sound like a negative position to take but its actually an honest and critical one. All software has bugs so a good tester will seek to find as many as he or she can.

## Inductive Reasoning Applied: Introduction

In lesson 1 we took at look at the basics of critical thinking and how we can apply the Socratic Method to form the right questions so we can solve particular problems. In this lesson, we're going to look closer at this method and particularly at how we can apply inductive reasoning to scientific problems.

Computer Science as a branch of engineering is a science in its own right--as the name implies. While writing computer programs or assembling computer hardware is a highly deductive process, you must use inductive reasoning to solve many problems prior to actually writing code or assembling a piece of hardware. We saw an example of this in Lesson 1 of this module but let's revisit some ways that inductive reasoning can be applied in the science.

1. **Forming a hypothesis**. Before a single line of code can be written or the first circuit board can be constructed, engineers are typically presented with a scenario to address, a problem to fix or a puzzle to solve. In order to make the first steps towards a solution, you first have to form a hypothesis about what might be the right problem and so you can apply the right solution.
2. **Confirmation**. The next step is to confirm your hypothesis is correct or if it needs modification. Suppose you are presented with a scenario in which the traffic at a particular light is always backed up at certain times of the day and you are asked to write software that will address the issue. You might formulate a hypothesis that this particular traffic light is not timed correctly -- one direction may be green for too long making the other directly green for too short a time. Before you start writing code to address the issue, you might confirm this hypothesis by running some models and tests. In this case you may write a set of programs that model the timing at the lights and the cars that go through it and change the timings to see if it has any effect on the problem.
3. **Determining Cause and Effect**. One danger in confirming or disconfirming a hypothesis is confusing correlation with causation. Causation is the direct relationship of one event with another event. Banging your fist on the desk is a direct cause of the sound that is created. Correlation is when two things happen at the same time but the one is not related to the other in a causal way. It's important to determine causal relations because if you mistake correlation for causation, you may end up solving the wrong problem!

## Forms of Inductive Reasoning

Recall that an inductive argument aims to show that if its premises all are true, then its conclusion, although not certain to be true, is probably true. (Recall also that by probably we mean “better than 50 percent likely.”) If an inductive argument succeeds in this aim, we call it a strong inductive argument. A strong argument thus has this feature: if its premises all are true, then its conclusion is probable, although not certain. An inductive argument that is not strong is said to be a weak inductive argument. A weak inductive argument is therefore an inductive argument in which it is not the case that if its premises all are true, then its conclusion is probably true. Methodical and intuitive procedures exist for the evaluation of inductive arguments; however, inductive arguments can take many different forms and the procedures vary from one form of inductive argument to another. We have space to examine three common forms of inductive reasoning.

## Arguing by Analogy

An analogical argument begins by asserting that two things, call them “X” and “Y,” have many features in common. This establishes an analogy (similarity) between X and Y. Next, an additional feature of X is identified—something not known to not be possessed by Y. The conclusion is that Y probably has this feature as well. For instance:

1. Monkeys and humans have similar hearts.
2. Drug X cures heart disease in monkeys.
3. Therefore, drug X will likely cure heart disease in humans.

A great deal of medical research is based on analogical reasoning. We reason this way often in everyday life.

## Enumerative Induction

Imagine that a professional dog-trainer with six large dogs just moved into the house next door. For the first week, every night at 1:00 a.m. he lets his dogs out in the yard for exercise. Every time, they run around and bark like crazy. It is 1 am on the eighth night and you naturally expect to hear dogs barking any minute. Your reasoning is as follows:

1. The dogs barked at 1 a.m. the first night.
2. The dogs barked at 1:00 a.m. the second night. (and so forth)
3. Therefore, they will probably bark tonight.

Your reasoning has taken the form of an enumerative induction. In such an argument, the premises list or enumerate information about a series of individuals or cases; a pattern is observed; and then a conclusion is drawn that extends that pattern to the next individual case, or to a whole group of like things. Essentially, this kind of inductive argument begins with a series of observations or cases and then extends the series to one or more new (unobserved) cases. The reasoning is inductive because the argument aims to show only that the conclusion is probable, not that the conclusion is completely certain.

## Inference to the Best Explanation

An inference to the best explanation occurs when we begin with one or more facts in need of explanation and then decide what to infer by thinking about what would best explain those facts. More formally, an inference to the best explanation typically takes the following form:

1. Begin with one or more facts in need of explanation.
2. Critically examine as many initially plausible potential explanations as possible (where a potential explanation is one that, if true, would explain the facts in question, and a plausible explanation is one that is consistent with general background information).
3. Rank one explanation as the best explanation on the basis of the standard criteria.
4. Conclude that the explanation ranked as the best explanation is probably true.

## The Logic of Science

Science operates on forming hypotheses and then spending time confirming or disconfirming those hypotheses. In this section we're going to take a look at a methodology--something you can apply to the problems you need to solve--that use inductive reasoning to form and confirm hypotheses. Applying the principles in this section can help you become a better engineer and even help you solve problems in life in general.

## The Hypothetico-Deductive Method

The method used to test an individual hypothesis in science is sometimes called the **hypothetico-deductive method** since it involves hypotheses and predictions deduced from them. The method is often presented in the following idealized steps:

**Step 1**. Scientists encounter a puzzling phenomenon that needs an explanation.

**Step 2**. Scientists propose a hypothesis that would, if true, explain the phenomenon.

**Step 3**. Scientists ask, If the hypothesis is true, what facts about the world can we expect to observe or otherwise detect with our senses? To answer this question, they derive **observational predictions** from the hypothesis.

**Step 4**. Scientists test the hypothesis by observing the world to see if the phenomenon predicted will be observed.

**Step 5**. Finally, they accept, reject, or revise the hypothesis on the basis of the test observations.

## Formulating a Hypothesis

Some people suppose that a hypothesis is derived straight from the data collected via a process of enumerative induction. Although this is sometimes the case, in most cases the hypothesis is “thought up” via a creative, imaginative process that is more like painting a picture than like completing a list (enumeration) of observations. Formulating an explanatory hypothesis after puzzling over the facts is a creative process requiring guesswork, inspiration, and imagination, combined with a basic knowledge of how the world works. No hard-and-fast rules tell the scientist how to come up with a good hypothesis.

## Deriving Predictions

In order for a hypothesis to count as scientific, it must be empirical in character. This rule has been part of the scientific enterprise since the beginnings of modern science in the sixteenth century. An empirical hypothesis is one that generates observational predictions—predictions that can be verified on the basis of observable facts (i.e., facts that can be detected using our physical senses). When scientists derive predictions from an empirical hypothesis, either deductive or inductive reasoning may be employed. Thus, deduction and induction are both used in science to derive predictions from hypotheses.

## Testing

The fact that some scientific hypotheses are literally “dreamed up” does not mean that scientists invent their own truths. The fact that a hypothesis has been put forward does not mean that it is true. Regardless of how a hypothesis was initially thought up, once it is presented to the scientific community, it must “run the gantlet.” The gantlet, of course, will be a series of standard scientific tests. A hypothesis will not be considered true until it has passed the standard tests. In addition, the tests will be performed by many researchers from various locations, and many of these researchers will have no vested interest in the hypothesis being found true. Indeed, some may actively try to disprove the hypothesis.

One important requirement at this stage of the investigation is reproducibility. The results of a test must be reproducible by any scientist anywhere, as long as the specified procedures are followed. There is never any guarantee that a hypothesis, once formed, will prove true.

## The Logic of Science: Confirmation and Disconfirmation

If the predicted phenomenon is observed, we say that the test "confirms the hypothesis." If the predicted phenomenon is not observed, we say the test "disconfirms the hypothesis." These two procedures, the **process of confirmation** and the **process of disconfirmation**, lie at the heart of the scientific method. The logic of confirmation takes the following form:

1. If the hypothesis is true, then P should be observed.
2. P is observed.
3. Therefore the hypothesis is probably true.

The logic of disconfirmation takes the following form:

1. If the hypothesis is true, then P should be observed.
2. P is not observed.
3. Therefore, the hypothesis is probably not true.

The more times a hypothesis has been confirmed, the higher the probability it is true. Thus, confirmation is never an all-or-nothing affair, and a hypothesis is never guaranteed to be true. Confirmation is always a matter of degree. Likewise, the more times a hypothesis has been disconfirmed, the more likely it is false. The evidence makes it probable to some degree that the hypothesis is false. (For a real-world example of this process at work, consult Dr. Herrick's book, Chapter 11).

## Comparing Rival Hypotheses

The hypothetico-deductive method is not the totality of the scientific method. It is an idealized process for testing one individual hypothesis, or for testing hypotheses one at a time. In most test situations, many different hypotheses will have been proposed to explain a given phenomenon, and each will have some explanatory power. Scientists will have to decide which of the rival hypotheses is the overall best explanation (and thus most likely to be true). How do scientists decide between competing hypotheses of a given phenomenon? They rely on the eight criteria for inference to the best explanation that we examined in the preceding chapter—essentially the same criteria we use in everyday life when we rank competing explanations, albeit with a few refinements. Einstein was right: science is indeed just a “refinement of everyday thinking”—a refinement of inference to the best explanation. In the preceding chapter we examined these criteria for selecting the best explanation:

1. internal consistency
2. external consistency
3. testability
4. explanatory scope
5. explanatory power
6. simplicity
7. fruitfulness
8. conservatism

## Mill's Method

When scientists track down the causes of things, the procedures they use are often based on a set of principles first formulated by the British philosopher John Stuart Mill (1806–1873) in his System of Logic published in 1843. These principles, now known as “Mill’s methods,” state procedures for identifying probable causes. Most of the principles are common sense. However, before we survey Mill’s methods, we must first clarify the concept of a cause.

## Cause and Effect

One of the key objectives of causal investigation is to discover, for a specified effect, the conditions under which the effect will or will not occur. Likewise, a key question of causal reasoning is this: under which conditions will the effect occur and under which conditions will the effect be absent? Thus, philosophers have found it illuminating to analyze causes in terms of underlying, or antecedent, conditions, and specifically in terms of two types of antecedents, or underlying, conditions: necessary conditions and sufficient conditions. Recall from an earlier lesson that a **necessary condition** for something X is a condition that is required; without the necessary condition, X will not obtain. For example, oxygen is necessary for a fire to burn. A **sufficient condition** for something X is a condition that guarantees that X will obtain; the sufficient condition is all that is needed for the obtaining of X. For example, jumping in a lake is sufficient for being wet. Notice that although oxygen is necessary for a fire to burn, it is not sufficient; while jumping in a lake is sufficient for getting wet but is not a necessary condition for getting wet. Mill’s methods are commonsense guides for discovering causally necessary and sufficient conditions.

In the next sections, we'll look at the specifics of Mill's Method and how the method can be used to determine cause.

## Mill's Method of Agreement

Suppose four students eat lunch in the school cafeteria, and later in the afternoon all four become sick. What is the cause of the illness? The health department would probably begin its investigation by asking each student what he or she ate for lunch. Suppose the information is compiled on the following table:

|  | **Fries** | **Salad** | **Fish Sticks** | **Burger** | **Soup** |
| --- | --- | --- | --- | --- | --- |
| Student 1 | X | X | -- | -- | X |
| Student 2 | -- | X | X | -- | -- |
| Student 3 | X | X | -- | -- | X |
| Student 4 | -- | X | X | X | -- |

Notice that each ate different items, with one exception: the salad. Given that the salad was the only factor all four cases had in common, the conclusion is that the salad is the probable cause of the illness. In a case like this, the health department would probably take samples of the salad and test them in its laboratory. The logical process illustrated in this case is Mill's method of agreement.

The basic idea here is that the probable cause of the effect E (the sickness) is to be found in the one antecedent condition common to each case where the effect is present. Of course, it is possible that something else, in addition to the salad, contributed to the outbreak. The cause identified by the method of agreement is likely a necessary cause—a necessary condition for the effect under investigation.

The method of agreement requires that we draw up a list of possible causes. We use our background knowledge of cause-and-effect connections when we do this. Next we look for one antecedent causal factor common to all cases of the effect in question. This common factor, if found, is identified as the probable necessary condition or as part of the necessary condition. Note that the conclusion is not that the condition singled out must be the cause; the conclusion is only that this is probably the cause (or is part of the cause).

## Mill's Method of Difference

Suppose Jan and Pat both have lunch at the school cafeteria. Later, in biology class, Jan gets sick, but Pat feels fine. The health department is again called in to investigate. Let us imagine that the information is compiled on the following table:

|  | **Salad** | **Burrito** | **Fried Rice** | **Burger** | **Sick after Lunch** |
| --- | --- | --- | --- | --- | --- |
| Jan | X | X | X | X | X |
| Pat | X | X | X | -- | -- |

The only difference between their lunches was that Jan ate the burger and Pat skipped the burger plate. Otherwise, they ate the same food at lunch. The burger is the obvious suspect. Why? It is the one factor present when the effect (the sickness) is present and absent when the effect is absent, where the two cases are otherwise similar. In such a case, it is natural to conclude that the burger probably caused the sickness.

This reasoning embodies Mill’s method of difference, which states that if an effect E is present in one case and absent in a closely similar case, the difference (D) between the cases is probably the cause of E. D will be the circumstance or condition present when E occurs and absent when E doesn’t occur—provided that the cases compared are otherwise alike in all, or nearly all, relevant respects. Unlike the method of agreement, the method of difference identifies a probable sufficient condition of an effect.

## Mill's Method of Concomitant Variation and Residues

## Mill's Method of Concomitant Variation

Mill’s method of concomitant variation looks for changes in one phenomenon that vary with or correspond to (are concomitant with) changes in a second phenomenon. If the measured change in the one varies along with the measured change in the second, this is evidence that the two phenomena are probably causally related: one of the two probably causes the other, or some third factor is the cause of both.

The method of concomitant variation is often used in everyday life as well as in the laboratory setting. For instance, suppose a community college district discovers a correlation between changes in enrollment and changes in employment at the local steel mill: when employment at the mill goes down, enrollment goes up; when employment at the mill goes up, enrollment goes down. The district officials conclude that employment at the mill is one of the causes of fluctuation in college enrollment.

However, a word of caution: the mere fact that two phenomena are correlated does not show with certainty that they are causally related. The number of Starbucks stores per square mile and the number of churches per square mile are statistically correlated. However, this does not prove that the presence of churches influences where Starbucks locates its stores. Common sense suggests that there is likely a third factor behind both sets of statistics, namely, population density.

## Mill's Method of Residues

The method of residues is common sense and takes this logical form.

1. A, B, and C are known to cause the set of effects X, Y and Z.
2. A is found to be the cause of X.
3. B is found to be the cause of Y.
4. So, C is likely the cause of the residue Z.

For example, a store owner discovers that three employees, A, B, and C, have embezzled a total of $1,500 from the tills. She then verifies that A embezzled $500 and that B embezzled $800. It follows that C embezzled the residual, $200.

## Inductive Reasoning: A Case Study

In this section, you're going to practice the principles you've been studying to a real-world scenario. This is a self-paced exercise that you will do on your own but we encourage you to post your response to the discussion board and get feedback so you can interact with your fellow students.

## Read

To begin read the article ["How to Help Self-Driving Cars Make Ethical Decisions"](https://www.technologyreview.com/s/539731/how-to-help-self-driving-cars-make-ethical-decisions/" \o "" \t "/home/ezequiel/Documentos\\x/_blank) by Will Knight on the MIT Technology Review. This article will present the scenario you will analyze using the tools from this lesson and other parts of the course.

Note: By clicking on this link, you will be taken to a third-party website. The inclusion of this link does not imply endorsement by Microsoft of the third-party, their website, or the materials, products and services available via the website. Microsoft has no responsibility or liability of any kind for the third-party website or the materials, products and services available via the website.

## The Problem Statement

You will use parts of the entire article to help you formulate the exact problem to solve but focus on this particular scenario in the article:

“When you ask a car to make a decision, you have an ethical dilemma,” says Adriano Alessandrini, a researcher working on automated vehicles at the University de Roma La Sapienza, in Italy. “You might see something in your path, and you decide to change lanes, and as you do, something else is in that lane. So this is an ethical dilemma.”

## The Exercise

To practice applying the skills taught in this lesson and throughout the course, do the following:

1. Clarify the problem statement. What problem or challenge is the researcher attempting to solve? Can you put the dilemma in the form of an inductive or deductive argument so you can figure out which premise or premises to test? Can you reword the scenario so it makes more sense to you?
2. Create an analogy. Come up with an analogy that you can use to clarify the problem. Start with "This problem is like . . ." and fill in the blank with something the dilemma is similar to. Describe why you think the analogy works. If you have trouble coming up with analogy, you may not fully understand the problem and may need to spend more time thinking about it.
3. Create scenarios that describe a solution and how to test those solutions.

* Now that you have clarity about what the problem is, how would you solve this using software? You don't need to write code of course but describe what the software might do. For example, you might talk about creating software that, "When the car senses two objects its path and cannot go in either direction, the car attempts to come to a complete stop." Of course that's one option but you also have to account for scenarios where that option isn't available and what might happen to the driver if the car came to a sudden stop from high speed (in other words, does your solution create a new problem?).
* Use Enumerative Induction to further describe your scenarios. How many tests would you need to determine that the scenario solves the dilemma. For example, how many times would the car coming to a complete stop and not hitting either object show that the objects are "safe"?
* Use the confirmation and disconfirmation method to test each scenario. For example:

Taking the scenario above, you can apply the confirmation method to it like this:

1. If the car coming to a complete stop would solve the dilemma is true, then neither object in either lane is struck by the car should be observed.
2. Neither object is struck by the car when it comes to a complete stop is observed.
3. Therefore the car coming to a complete stop would solves dilemma is probably true.

You can also apply the disconfirmation method in the same way.

1. What other tools from this lesson or elsewhere in the course can you use to better analyze the problem and come up with solutions?