# Chapter 5: Propositional Logic Translations[[1]](#footnote-1)

## I. Introduction

You have seen the basics of how arguments work and the fundamental building blocks of our logical thinking. The focus of this chapter is putting these building blocks together in a more formal fashion, and then using the language and operations of logic to create and analyze deductive arguments. The purpose of deductive logic is to figure out what new information we can obtain from information we already have, and propositional logic is a useful way of doing this.

### Why Propositional Logic?

The fundamental logical unit of the system you’ll be learning in this chapter is the proposition; therefore, we call it “propositional logic.” (Some refer to the same chunks of information as sentences, and call this system “sentential logic” instead[[2]](#footnote-2)). A proposition is the meaning behind the words; the same proposition can be expressed different ways. For example, “My brother is tall,” “My brother Wade is tall,” and “Wade is tall,” are different sentences, but express the same proposition.

In this system of logic, we’ll break complex sentences down into simple propositions and logical terms. This system has its roots in Ancient Greece, but it wasn’t until the 19th century that it was developed into a formal system. It is not the only system of symbolic logic that was developed in the 19th and 20th centuries, but it is the most direct and fundamental. Most systems of symbolic logic begin with propositional logic and develop it into other forms to suit other purposes, depending on the level of logical complexity we wish to analyze.

In this chapter, we will discuss the basics of the proposition-centered approach to deductive logic. Propositional logic must accomplish three tasks:

1. Tame natural language.

2. Precisely define logical forms.

3. Develop a way to analyze logical forms for validity.

The approach to the first task—taming natural language—will be accomplished by essentially translating natural language (in our case, English) into this symbolic system. A lot of the nuances of natural language are lost, but the logic itself becomes very clear, which is our main goal.

We will call our artificial language “PL,” short for ‘Propositional Logic.’ In constructing a language, we must specify its syntax and its semantics. By the *syntax* of a language, we mean the rules governing what counts as a well-formed construction within that language; that is, syntax is the language’s grammar. Syntax is what tells me that ‘What a handsome poodle you have there.’ is a well-formed English construction, while ‘Poodle a handsome there you what have.’ is not. So, the syntax of PL will tell you in what order we can put the letters, logical symbols, and parentheses so the resulting formula is well-formed, instead of logical nonsense. While learning the symbols and their syntax, we will also learn some rules for how to translate from English into PL. The *semantics* of a language is an account of the meanings of its well-formed bits. In PL, this is different from the correspondence between the symbols and English. Our semantics section will tell you under what conditions a given proposition is true or false.

## II. Syntax and Translation

First, we cover syntax. This discussion will give us some clues as to the relationship between PL and English, and it will teach us how to translate English sentences into PL. We can distinguish, in English, between two types of (declarative) propositions: simple and compound. A simple proposition is one that does not contain any other sentence as a component part. A compound proposition is one that contains at least one other sentence as a component part. Compound propositions take simple propositions and add logic, often (but not always) combining more than one simple proposition together. ‘Beyoncé is logical’ is a simple proposition; none of its parts is itself a sentence. ‘Beyoncé is logical and James Brown is alive’ is a compound proposition: it contains two simple sentences as component parts—namely, ‘Beyoncé is logical’ and ‘James Brown is alive’—and it combines them using the logical word ‘and.’

### Simple propositions, compound propositions

In PL, we will use capital letters—‘A’, ‘B’, ‘C’, …, ‘Z’—to stand for simple sentences. You can use any capital letters you want for your simple sentences, as long as you remember two rules: in any particular compound sentence, paragraph, or argument, (1) you cannot use the same letter to refer to different propositions, and (2) you have to make sure you use the same letter for a proposition every time it appears. If I chose the letter ‘B’ to refer to ‘Beyoncé is logical,” and later chose ‘B’ to refer to ‘James Brown is alive,’ that would be confusing; that’s the importance of rule (1). For rule (2), you need to make sure the same proposition gets the same letter, *even if we use different words to express it.* If I give a ‘J’ to ‘James Brown is alive,’ and later on in the same argument I state ‘he is alive,’ if it is clear the pronoun refers to James Brown, ‘he is alive’ will also get a ‘J.’

When we replace the simple propositions with letters, what ends up being left are logical terms. ‘Beyoncé is logical and James Brown is alive,’ using the letters we chose above, becomes ‘B and J.’ We are very soon going to replace the logical words with symbols, but I suggest you begin by simply replacing the simple propositions with letters, and retaining everything else – including all logical words and, for now, punctuation.

Here’s a more complex example:

Neither my brother nor sister are tall, unless both my mom and dad are tall.

This one has four different simple sentences: ‘my brother is tall,’ ‘my sister is tall,’ ‘my mom is tall,’ and ‘my dad is tall.’ Notice that I only said the word ‘tall’ twice in the original sentence, but I *meant* it four times—it is meant to apply to both of my siblings, and to both of my parents. We just don’t like repeating ourselves unnecessarily, so in natural language, we shorten things where we can.

Let’s assign letters to these four simple sentences:

B = My brother is tall.

S = My sister is tall.

M = My mom is tall.

D = My dad is tall.

Now, replacing these chunks of information with the letters, and keeping the logical words and punctuation, we end up with:

Neither B nor S, unless both M and D.

The next step in translation will be to replace the logical words with symbols. We’ll also use parentheses when we need to group things. You’ll notice in the example above that the comma separates ‘neither B nor S’ from ‘both M and D.’ We’re going to drop the comma, so we’ll use parentheses to let us know that the word ‘unless’ is the logical concept that brings the whole sentence together—it’s the main connective.

We will have five total logical symbols, each representing a different concept: conjunctions, disjunctions, negations, conditionals, and biconditionals.

### 1. Conjunctions

The first type of compound proposition is one that we’ve already seen. Conjunctions are, roughly, ‘and’ sentences. A conjunction takes two propositions and tells us both of them are true. For the conjunction “Beyoncé is logical and James Brown is alive,” we’ve already decided to let ‘B’ stand for “Beyoncé is logical” and to let ‘J’ stand for “James Brown is alive.” What we need is a symbol that stands for ‘and’. In PL, that symbol is a “dot.”. It looks like this: •. To form a conjunction in PL, we stick the dot between the two component letters:

B dot J

That is the PL version of “Beyoncé is logical and James Brown is alive.”

A note on terminology. The dot goes between the two things it connects, so a conjunction has two components, one on either side of the dot. We will refer to these as the “conjuncts” of the conjunction. If we need to be specific, we might refer to the “first conjunct” (‘B’ in this case) or the “second conjunct” (‘J’ in this case).

There are a lot of words in the English language that mean, logically, “and.” Here are a few of them:

|  |  |  |
| --- | --- | --- |
| and | both … and | also |
| moreover | however | but |
| although | furthermore | as well as |

You’ll notice these words and phrases have slightly different nuances. If I were to tell you “My brother and sister are tall,” and I wanted to use a different word for ‘and,’ only some of these would work. I would not say, for example, “My brother is tall but my sister is tall.” We use ‘but,’ ‘however,’ and ‘although’ when the second sentence is significantly different from the first. I would use it here: “My brother is tall but my sister is short.” Logically, though, all of these conjunctions are simply telling us “both of these things are true.”

### 2. Disjunctions

Disjunctions are, roughly, ‘or’ sentences. A disjunction takes two propositions and tells us that *at least one of them* is true. For example, “Beyoncé is logical or James Brown is alive.” Sometimes, the ‘or’ is accompanied by the word ‘either,’ as in “Either Beyoncé is logical or James Brown is alive.” Again, we let ‘B’ stand for “Beyoncé is logical” and let ‘J’ stand for “James Brown is alive.” What we need is a symbol that stands for ‘or’ (or ‘either/or’). In PL, that symbol is a “wedge.” It looks like this: v.

To form a disjunction in PL, we simply stick the wedge between the two component letters, thus:

B wedge J

That is the PL version of “Beyoncé is logical or James Brown is alive.” It is also the way we symbolize “Either Beyoncé is logical or James Brown is alive.” We don’t need a separate symbol for the ‘either’ here—though it is a useful word, as we can use it to figure out how a sentence is grouped, if we need to use parentheses in our translation.

There are many, many words that are conjunctions, but only a few that are true disjunctions.

|  |  |  |
| --- | --- | --- |
| or | either … or | unless |

The last one is a weird one; you might not expect it to be equivalent to an ‘or’ statement, but it turns out that it is.[[3]](#footnote-3)

A note on terminology. A disjunction has two components, one on either side of the wedge. We will refer to these as the “disjuncts” of the disjunction. If we need to be specific, we might refer to the “first disjunct” (‘B’ in this case) or the “second disjunct” (‘J’ in this case).

### 3. Negations

Negations are, roughly, ‘not’ sentences—sentences like “James Brown is not alive.” A negation takes one proposition and tells us that it’s false. You may find it surprising that this would be considered a compound sentence. Remember that a simple proposition is one that does not contain any other propositions within it. “James Brown is not alive” does contain another proposition inside it: “James Brown is alive.” Another way to think of it is that “James Brown is not alive” is a compound proposition because it takes a simple proposition and combines it with a logical concept. Any time we add logic, we’re forming compound proposition. We have ‘J’ to stand for the simple proposition; we need a symbol for ‘it is not the case that.’ In PL, that symbol is a “tilde.” It looks like this: ~.

To form a negation in PL, we simply prefix a tilde to the simpler component being negated:

tilde JThis is the PL version of “James Brown is not alive.”

There are many, many words and phrases that are negations. Here are a few:

|  |  |  |
| --- | --- | --- |
| not | it is false that | it is not the case that |
| never | nowhere | nothing |
| no one | neither … nor | n’t |

That last one, ‘n’t,’ usually sticks onto another word: isn’t, wasn’t, won’t. Actually, the last two rows have negations that are stuck onto another word. ‘Isn’t’ is short for ‘is not.’ ‘Never’ is short for ‘not ever.’ ‘Nothing’ is short for ‘not anything’ or ‘not something.’ Importantly for us, ‘neither … nor’ is short for ‘not either … or.’ That’s both a disjunction and a negation. Watch out for negations; they hide. Sometimes, we don’t even express the negation: “James Brown is not alive” means the same thing as “James Brown is dead.” If you’ve already given a letter to “James Brown is alive,” then both of these should be translated the same way – with a tilde.

### 4. Conditionals

Conditionals are, roughly, ‘if … then’ sentences—sentences like “If Beyoncé is logical, then James Brown is alive.” Again, we let ‘B’ stand for “Beyoncé is logical” and let ‘J’ stand for “James Brown is alive.” What we need is a symbol that stands for the ‘if/then’ part. In PL, that symbol is a “horseshoe.” It looks like this: ⊃.

To form a conditional in PL, we simply stick the horseshoe between the two component letters (where the word ‘then’ occurs), thus:

B horseshoe JThat is the PL version of “If Beyoncé is logical, then James Brown is alive.”

Conditionals are tricky, for two reasons. First, there are again a lot of words and phrases that express conditionals. Second, order matters, and humans do not always present their conditionals in if … then order. Imagine I give my child a promise: if she eats her veggies, then she’ll eat dessert. It means something different if I switch the order and say “If you eat dessert, then you’ll eat your veggies.” The dessert is supposed to be her reward, the consequence of her eating her veggies as asked. But I might switch the order I express them in, and say something like “You can have dessert, provided that you eat your veggies,” dangling the reward first, and then telling her the condition she must meet to earn it. Since the horseshoe always means ‘if … then’ in that order, I need to switch the two sides here so the veggies come before the horseshoe, and dessert comes last.

Here are some words and phrases that should be horseshoes, and you get to keep them in the same order:

|  |  |
| --- | --- |
| If A then B  If A, B  A is a sufficient condition for B | A implies B  A only if B |

All of these should be translated as:

A horseshoe BAnd here are some words and phrases that should be horseshoes, but you need to switch the order:

|  |  |
| --- | --- |
| A, if B | A provided that B |
| A on the condition that B | A given that B |
| A is a necessary condition for B |  |

All of these should be translated as:

B horseshoe AA note on terminology. Unlike our treatment of conjunctions and disjunctions, we will distinguish between the two components of the conditional, since order matters. The component that comes before the horseshoe will be called the “antecedent” of the conditional; the component after the horseshoe is its “consequent.”

### 5. Biconditionals

Biconditionals are, roughly, ‘if and only if’ sentences—sentences like “Beyoncé is logical if and only if James Brown is alive.” Again, we let ‘B’ stand for “Beyoncé is logical” and let ‘J’ stand for “James Brown is alive.” What we need is a symbol that stands for the ‘if and only if’ part. In PL, that symbol is a “triple-bar.” It looks like this: ≡.

To form a biconditional in PL, we simply stick the triple-bar between the two component letters, thus:

B triple-bar JThat is the PL version of “Beyoncé is logical if and only if James Brown is alive.”

There are very few phrases that translate as biconditionals. We’ve got three:

|  |  |  |
| --- | --- | --- |
| if and only if | necessary and sufficient condition | is logically equivalent to |

‘If’ by itself is a conditional and gets a horseshoe; ‘only if’ by itself is a conditional. ‘If and only if’ is a biconditional. A biconditional has two conditionals, like a bicycle has two wheels. Similarly, ‘necessary’ by itself is a conditional; ‘sufficient’ by itself is a conditional. ‘Necessary and sufficient condition’ is a biconditional.

There are no special names for the components of the biconditional, and order does not matter.

### Punctuation: Parentheses and Brackets

Our language, PL, is quite austere: so far, we have only 31 different symbols—the 26 capital letters, and the five symbols for the five different types of compound sentence. We will now add parentheses and brackets. And that’ll be it.

We use parentheses in PL for one reason (and one reason only): to remove ambiguity. To see how this works, it will be helpful to draw an analogy between PL and the language of simple arithmetic. The latter has a limited number of symbols as well: numbers, signs for the arithmetical operations (addition, subtraction, multiplication, division), and parentheses. The parentheses are used in arithmetic for disambiguation. Consider this combination of symbols:

two plus three times fiveAs it stands, this formula is ambiguous. If I’ve forgotten my order of operations, I don’t know whether this is a sum or a product; that is, I don’t know which operator—the addition sign or the multiplication sign—is the main operator. We can use parentheses to disambiguate, and we can do so in two different ways:

parenthesis two plus three end parenthesis times fiveor

two plus parenthesis three times five end parenthesisAnd of course, where we put the parentheses makes a big difference. The first formula is a product; the multiplication sign is the main operator. It comes out to 25. The second formula is a sum; the addition sign is the main operator. It comes out to 17. Different placement of parentheses, different results.

This same sort of thing is going to arise in PL. Our logical operators are the dot, wedge, tilde, horseshoe, and triple-bar, and we need to be able to tell what’s the main operator. There are ways of combining PL symbols into compound formulas with more than one operator; and just as is the case in arithmetic, without parentheses, these formulas would be ambiguous. Let’s look at an example.

Consider this sentence: “If Beyoncé is logical and James Brown is alive, then I’m the Queen of England.” This is a compound proposition, but it contains both the word ‘and’ and the ‘if … then’ construction. It has three simple components: the two that we’re used to by now about Beyoncé and James Brown, which we’ve been symbolizing with ‘B’ and ‘J,’ respectively, and a new one—“I’m the Queen of England”—which we may as well symbolize with a ‘Q.’ Based on what we already know about how PL symbols work, we would render the sentence like this:

B dot J horseshoe QBut just as was the case with the arithmetical example above, this formula is ambiguous. I don’t know what kind of compound proposition this is, a conjunction or a conditional. That is, I don’t know which of the two operators—the dot or the horseshoe—is the main operator. In order to disambiguate, we need to add some parentheses. There are two ways this can go, and we need to decide which of the two options correctly captures the meaning of the original sentence:

Parenthesis B dot J end parenthesis horseshoe Qor

B dot parenthesis J horseshoe Q end parenthesis

The first formula is a conditional; horseshoe is its main operator, and its antecedent is a compound sentence. The second formula is a conjunction; dot is its main operator, and its second conjunct is a compound sentence. We need to decide which of these two formulations correctly captures the meaning of the English sentence “If Beyoncé is logical and James Brown is alive, then I’m the Queen of England.”

We have two clues that the first translation, that groups B and J together, is the correct one. First, we’ve expressed our conditional with two words: if … then. Everything between the ‘if’ and the ‘then’ is the antecedent. B and J come between ‘if’ and ‘then,’ so they’re grouped together as the antecedent. The second clue is that comma—remember I told you to pay attention to punctuation? A comma often tells you where the sentence splits. In this case, it separates B and J from the Q. If a comma is the strongest form of punctuation inside a sentence, you will notice that often, the logical word or phrase right after the comma is the main operator. So, the correct translation is:

Parenthesis B dot J end parenthesis horseshoe Q

Again, in PL, parentheses have one purpose: to remove ambiguity. We only use them for that. This kind of ambiguity arises in formulas, like the one just discussed, involving multiple instances of the operators dot, wedge, horseshoe, and triple-bar.

Let’s translate a more complex sentence—the one I gave you above, before we started learning the logical symbols. Remember it was:

Neither my brother nor sister are tall, unless both my mom and dad are tall.

See that comma in the middle? That splits our sentence into “Neither my brother nor sister are tall” on the one side and “Both my mom and dad are tall” on the other. ‘Unless’ is the word that will end up being the main connective. Let’s take one side at a time. Remember that ‘neither … nor’ is short for ‘not either … or’—it’s both a negation and a disjunction. More specifically, it’s the negation OF a disjunction. So, I want to use both a tilde and a wedge. Here’s my first (incorrect) try:

tilde B wedge S

What’s wrong with this? Tildes are sticky—they stick to whatever comes directly after them and negate only that. What I’ve ended up saying here is “Either my brother is not tall, or my sister is tall.” That’s not what the original said. Neither B nor S means the entire disjunction, B or S, is false. So, I need to collect that disjunction in parentheses, and then stick the tilde on the front of that, like this:

tilde parenthesis B wedge S end parentesis

Now let’s tackle the second side: “Both my mom and dad are tall.” That’s easy; that’s just a dot put between M and D. Now, if that was a stand-alone sentence, that’s all we have to do, but I want to attach this compound sentence to another sentence, so I need to group it in parentheses:

parenthesis M dot D end parenthesis

Last thing! I need to connect these two sides together. The word used here is ‘unless.’ Remember, ‘unless’ should be translated like a disjunction. The symbol for disjunction is the wedge, and our final product is:

tilde parenthesis B wedge S end parenthesis wedge parenthesis M dot D end parenthesis

Before moving on, I want to say a quick word about “neither … nor.” You’ll notice we put the tilde on front of parentheses, with the wedge in the middle—it is not the case that (B or S). We know we can’t just stick the tilde to the B, that says something different. But can’t we just put a tilde in front of both B and S, ending up with:

tilde B wedge tilde S

NO YOU CAN NOT. (Sorry for the all-caps, I really want to emphasize that). This says “Either my brother is not tall, or my sister is not tall.” That means something different. Think about it. If I tell you “Neither my brother nor sister are tall,” how many tall siblings do I have? Zero. If I tell you “Either my brother is not tall, or my sister is not tall,” how many tall siblings do I have? Well, I don’t know. My brother could be the short one, my sister could be the short one, or both of them could be short. I can have one tall sibling and make that ‘either … or’ sentence true—it just tells me someone is not tall, but doesn’t tell me who. The ‘neither … nor’ sentence is more specific. I know exactly who is tall: no one.

In addition to parentheses, you’ll also see (and use) brackets. They mean exactly the same thing as parentheses do—they just group things together so you can find the main operative of the sentence, and the main operative of each part of the sentence. We use them when we need to group something that already has used parentheses. For example, consider this:

parenthesis A wedge B end parenthesis, dot, parenthesis C horseshoe D end parenthesisNo, I don’t know what it means in English, but I do know it’s a conjunction—the main operator is the dot, and the dot connects two compound statements to each other. Suppose, though, that this whole conjunction is false? I need to add a tilde, but if I just put it on the front without further grouping, it sticks to that first parenthesis, and negates just the first half of the sentence, instead of the whole thing. I need to group the entire sentence together and put a tilde on the front of that. Since this sentence already has parentheses, I’m going to step it up to square brackets, just so it’s easier to see how things are grouped:

tilde bracket parenthesis A wedge B end parenthesis, dot, parenthesis C horseshoe D end parenthesis end bracket

### Translation Tips

Here’s some advice for how to approach translating sentences from English into PL.

#### Assigning letters to propositions

You need to start with this step, assigning letters to propositions. When you’re translating more complicated sentences into PL, take the time to rewrite the sentence using the letters instead of the propositions they represent. Make sure you keep all logical words and punctuation. So, if you see a sentence like this:

The Mandalorian will take his helmet off in the Grogu movie only if Pedro Pascal finishes filming Fantastic Four in time; but him taking his helmet off is a necessary condition for getting a lot of Pascal’s fans to see it in the theater, and a lot of his fans seeing it in the theater is sufficient for the movie to become a huge success.

Take the time to rewrite it like this:

M only if P; but M is a necessary condition for F, and F is sufficient for S.

Now that you’ve gotten rid of the content and retained the logic, you can stop thinking about Pedro Pascal, and you can focus on picking the right connectives and figuring out where the parentheses are going to go.

Next tip: make sure you give capital letters to *full* propositions. “Neil Armstrong walked on the moon” should be given a single letter representing the whole sentence. A lot of students have the strong impulse to give letters to all noun terms and have something like “NA walked on M.” The whole thing just gets a single letter, so this is just translated as “N.”

Only give capital letters to *simple* propositions. Look especially for negations, and extract them as a logical symbol. If I see “My brother is not tall,” I want to give the letter to the positive simple proposition “My brother is tall” and add the negation back later as a logical symbol.

Don’t use the same letters for different simple propositions. We can get in the habit of picking a letter related to any names we see in the sentence, like I did above with Neil Armstrong. But if you get a sentences like this, “Jodie went to the movies but Johnny did not,” it’s tempting to give both propositions a ‘J.’ If we use ‘J’ for Jodie, we need to pick a different letter for Johnny’s movie-going habits.

Make sure you use the same letter for the same proposition. Look for the meaning of the sentence; if it means the same as a previous sentence, even though it’s worded differently, it needs to get the same letter. In the Mando example above, I gave the M to both “The Mandalorian will take his helmet off in the Grogu movie,” and to “him taking his helmet off.” We understand from context that the second string of words means the same thing as the first, so they get the same letter.

#### Picking the right connective

Keep in mind what each symbol means. The words representing each connective I’ve listed above are only a partial list, especially for negations, conjunctions, and conditionals. So, ask yourself:

* Are they trying to tell you that both things are true? That’s a conjunction and gets a dot.
* Are they saying at least one of these things are true? That’s a disjunction and gets the wedge.
* Are they saying something is false? That’s a negation and gets a tilde.
* Are they pointing out a connection between two things, such as a promise, or a causal connection? That’s a conditional and gets a horseshoe UNLESS…
* Is it a *very* strong connection? That’s probably a biconditional, and gets a triple-bar.

#### Some advice on conditionals

Every time you see a conditional, STOP and ask yourself if you can keep the two terms in the same order or if you need to switch them. Re-read the section above on conditionals several times until you get a good feel for it. The Mandalorian example above has three conditionals:

M only if P; but M is a necessary condition for F, and F is sufficient for S.

Remember, “only if” and “sufficient for” do not switch order. But if you have “necessary condition for,” then you have to switch the order. So the three conditionals will be translated as:

#### M horseshoe P. F horseshoe M. F horseshoe S.

#### Grouping

Remember, we need to use parentheses so we can find the main operator of each sentence, and the main operator of each part of the sentence. A good rule of thumb is that if you are doing logic on a compound statement, you’ll need to group that statement into parentheses or brackets first. So, negating a compound statement means you put the statement in parentheses, and then put the tilde on front. Hooking two compound statements together means each of them needs to be grouped before you connect them.

Figuring out how to group things is more of an art form than a science; you really have to read the sentences carefully to figure it out. Look for clues in the language and punctuation. Commas and semi-colons give you clues of how to group things. How we express ourselves also gives clues to how we think things should be grouped. For example, look at these two sentences:

1. My mom and dad will go to dinner unless my uncle does.
2. My mom is going to dinner, and my dad will go unless my uncle does.

Without parentheses, both of these look the same. Giving ‘M’ to the proposition “My mom is going to dinner,” ‘D’ to “my dad is going to dinner,” and ‘U’ to “my uncle is going to dinner,” they both look like this, pre-grouping:

M dot D wedge U

Yet they mean different things. #1 means if my uncle does not go, then both my mom dad will go—their dinner plans depend on what my uncle does. #2 means my mom is going for sure, but my dad’s going depends on what my uncle does. #2 has that nice comma telling us where the sentence breaks. For #1, we focus on how we’ve grouped mom and dad together linguistically. Instead of “My mom is going to dinner and my dad is going to dinner,” we’ve got “my mom and dad are going to dinner.” These two simple propositions are grouped so closely we’ve smushed their sentences together. So, the translations for these two should be:

1. Parenthesis M dot D end parenthesis wedge U. 2. M dot parenthesis D wedge U end parenthesis.

And, just because I can’t leave an example unfinished, let’s finish translating the Mandalorian sentence. Here’s a reminder of the sentence we’re working with:

M only if P; but M is a necessary condition for F, and F is sufficient for S.

We figured out that the first and third conditionals need to keep the letters in this same order, while the middle conditional switches the order. We know how things are grouped because of the comma and semi-colon. The semi-colon is stronger than the comma, so that tells me the whole sentence breaks there, and the last two conditionals are grouped together. And finally, both “but” and “and” are conjunctions, so they get dots. Here’s the complete translation:

Parenthesis M horseshoe P end parenthesis, dot, bracket parenthesis F horseshoe M end parenthesis dot parenthesis F horseshoe S end parenthesis end bracket.

I had to use square brackets, because that side of the sentence groups two compound sentences together; since I’d already used parentheses, the square brackets help me keep track of how the whole sentence comes together.

Now it’s your turn! Good luck!

## III. Semantics of Propositional Logic

Our task is to give precise meanings to all of the well-formed formulas of PL. Some of this task is already complete. We know something about the meanings to the 26 capital letters: they stand for simple English sentences of our choosing. We know roughly how each symbol corresponds to English logical words and phrases. We need more precise semantics, however. We want to know what makes a compound sentence *true*. We need the truth conditions for our operators.

A sentence in PL can have one of two semantic values: true or false. That’s it. This is one of the ways in which the move to PL is a taming of natural language. In PL, every sentence has a determinate truth-value; and there are only two choices: true or false. English and other natural languages are more complicated than this. Of course, there’s the issue of nondeclarative sentences (such as questions or commands), which don’t express propositions and don’t have truth-values at all. But even if we restrict ourselves to declarative English sentences, things don’t look quite as simple as they are in PL. Consider the sentence “Napoleon was short.” You may not be aware that the popular conception of the French Emperor as diminutive in stature has its roots in British propaganda at the time. As a matter of fact, he was about 5’ 7”. Is that short? Well, not at the time (late 18th, early 19th centuries); Napoleon was about average or slightly above for that time period. People are taller now, though, so is 5’ 7” short from today’s perspective? The average height for a modern Frenchman is 5’ 9.25”. Napoleon is 2.25 inches shorter than average. How much shorter than average do you have to be to qualify as ‘short?’ Heck, I don’t know!

The problem here is that relative terms like ‘short’ have borderline cases; they’re vague. It’s not clear how to assign a truth-value to sentences like “Napoleon was short.’ So, in English, we might say that they lack a truth-value (at least until we define our terms more precisely). Some systems of logic that are more sophisticated than our PL have developed ways to deal with these sorts of cases. Instead of just two truth-values, some systems add more. There are three-value systems, where you have true, false, and neither. There are systems with infinitely many truth-values between true and false (where false is zero and true is 1, and every real number in between is a degree of truth). The point is, English and other natural languages are messy when it comes to truth-value. We’re taming them in PL by assuming that every PL sentence has a determinate truth-value, and that there are only two truth-values: true and false—which we will indicate, by the way, with the letters ‘T’ and ‘F.’

Our task here is to provide truth conditions for the five operators: dot, wedge, tilde, horse shoe, and triple-bar, and horseshoe (we start with the dot because it’s the most intuitive). We will specify the meanings of these symbols in terms of their effects on truth-value: what is the truth-value of a compound sentence featuring them as the main operator, given the truth-values of the components? The semantic values of the operators will be truth functions: systematic accounts of the truth-value outputs (of the compound proposition) resulting from the possible truth-value inputs (of the simpler components). Another way to put this—we will specify the truth conditions for each connective; under what conditions would this compound sentence be true, and under what conditions would it be false?

### 1. Conjunctions (DOT)

Our rough-and-ready characterization of conjunctions was that they are ‘and’ sentences— sentences like “Beyoncé is logical and James Brown is alive.” Since these sorts of compound sentences involve two simpler components, we say that dot is a two-place operator. It takes two sentences and hooks them together. So, when we specify the general form of a conjunction using generic variables, we need two of them. The general form of a conjunction in PL is:

p dot qWe’re using lower-case p and q here, because capital letters represent specific simple sentences. Here we want to talk about any sentence that has a dot as a main connective, so p and q are variables which could be replaced with any sentence, simple or compound.

To figure out the truth conditions of a conjunction, the questions we need to answer are these: Under what circumstances is the entire conjunction true, and under what circumstances false? And how does this depend on the truth-values of the component parts? We remarked earlier that when someone utters a conjunction, they’re committing themselves to both of the conjuncts—they’re telling you both conjuncts are true. If I tell you “Beyoncé is wise and James Brown is alive,” I’m committing myself to the truth of both of those alleged facts; so, if even one of them turns out false, I’ve have not told you the truth.

This is how conjunctions work, then: they’re true just in case both conjuncts are true; false otherwise. We can represent this graphically, using what we’ll call a “truth-table”:

|  |  |  |
| --- | --- | --- |
| p | q | p • q |
| T | T | T |
| T | F | F |
| F | T | F |
| F | F | F |

Since the dot is a two-place operator, we need columns for each of the two variables in its general form—p and q. Each of these is a generic PL sentence that can be either true or false. That gives us four possibilities for their truth-values as a pair: both are true, p is true but q is false, p is false but q is true, or both false. These four possibilities give us the four rows of the table. For each of these possible inputs to the truth-function, we get an output, listed under the dot. T is the output when both inputs are Ts; F is the output in every other circumstance.

In other words, a conjunction is only true when both sides are true, and false in every other circumstance, which is exactly what we know ‘and’ statements mean. If I told you my brother and sister are tall, you would expect me to have two tall siblings.

### 2. Disjunctions (WEDGE)

Our rough characterization of disjunctions was that they are ‘or’ sentences—sentences like “Beyoncé is logical or James Brown is alive.” In PL, the general form of a disjunction is:

p wedge q

where p and q are variables representing any sentence in PL, simple or compound.

We need to figure out the circumstances in which such a ‘or’ statements are true; we need the truth-function represented by the wedge. While ‘and’ statements have only one way of making them true, ‘or’ statements have more. If I tell you either my brother or sister are tall, I’ve told you someone is tall, but I haven’t told you who.

At this point we face a complication. Wedge is supposed to capture the essence of ‘or’ in English, but the word ‘or’ has two distinct senses. This is one of those cases where natural language needs to be tamed: our wedge can only have one meaning, so we need to choose between the two alternative senses of the English word ‘or.’

‘Or’ can be used *exclusively* or *inclusively*. The exclusive sense of ‘or’ is expressed in a sentence like this: “Candidate A will win the election, or Candidate B will win.” The two disjuncts present exclusive possibilities: one or the other will happen, but not both. The inclusive sense of ‘or,’ however, allows the possibility of both. If I told you I was having trouble deciding what to order at a restaurant, and said, “I’ll order lobster or steak,” and then I ended up deciding to get both, you wouldn’t say I had lied to you when I said I’d order lobster or steak. The inclusive sense of ‘or’ allows for one or the other—or both.

We will use the inclusive sense of ‘or’ for our wedge. There are arguments for choosing the inclusive sense over the exclusive one, but we will not dwell on those here. As we will see later, the exclusive sense will not be lost to us because of this choice: we will be able to symbolize exclusive ‘or’ within PL, using a combination of operators.

So, wedge is an inclusive disjunction. It’s true whenever one or the other—or both—conjuncts is true; false otherwise. This is its truth-table definition:

|  |  |  |
| --- | --- | --- |
| p | q | p v q |
| T | T | T |
| T | F | T |
| F | T | T |
| F | F | F |

If there are any trues at all, a disjunction is true; it is only false where both disjuncts are false. If I told you either my brother or sister are tall, that sentence is true if the tall one is my brother, my sister, or both of them.

### 3. Negations (TILDE)

Tilde is a one-place operator. It does not connect two sentences together, like two-place operators; instead, it attaches to just one sentence. The general form of a negation is:

tilde pwhere ‘p’ is a variable standing for any generic PL sentence, simple or compound.

We need to give an account of the meaning of the tilde in terms of its effect on truth-value. Tilde, as we said, is the PL equivalent of ‘not’ or ‘it is not the case that.’ Let’s think about what happens in English when we use those terms. If we take a true sentence, say “Edison invented the light bulb,” and form a compound with it and ‘not,’ we get “Edison did not invent the light bulb”—a falsehood.[[4]](#footnote-4) If we take a false sentence, like “James Brown is alive,” and negate it, we get “James Brown is not alive”—a truth.

Evidently, the effect of negation on truth-value is to turn a truth into a falsehood, and a falsehood into a truth. We can represent this graphically, using what we’ll call a “truth-table.” The following table gives a complete specification of the semantics of tilde:

|  |  |
| --- | --- |
| p | ~ p |
| T | F |
| F | T |

Because tilde is a one-place operator, our table only needs two lines to represent all possible truth values: a single sentence in PL is either true or false, those are all of your options. We can compute the truth-value of the negation based on the truth-value of the sentence being negated: if the original sentence is true, then its negation is false; if the original sentence is false, then the negation is true. The logical function of negations is to flip the truth value of whatever sentence to which they attach.

### 4. Conditionals (HORSESHOE)

Our rough characterization of conditionals was that they are ‘if … then’ sentences—sentences like “If Beyoncé is logical, then James Brown is alive.’ We use such sentences all the time in everyday speech, but is surprisingly difficult to pin down the precise meaning of the conditional, especially within the constraints imposed by PL. There are in fact many competing accounts of the conditional—many different conditionals to choose from—in a literature dating back all the way to the Stoics of ancient Greece. Whole books can be, and have been, written on the topic of conditionals. In the course of our discussion of the semantics for horseshoe, we will get a sense of why this is such a vexed topic; it’s complicated.

The general form of a conditional in PL is:

p horseshoe q

We need to decide for which values of p and q the conditional turns out true and false. To help us along let’s consider a conditional claim with a little story to go along with it. Suppose Barb is suffering from joint pain; she doesn’t know what is causing it and hasn’t been to the doctor to find out. She’s complaining about her pain to her neighbor, Sally. After hearing a brief description of the symptoms, Sally is ready with a prescription, which she delivers to Barb in the form of a conditional claim: “If you drink this herbal tea every day for a week, then your pain will go away.” She hands over a packet of tea leaves and instructs Barb in their proper preparation.

We want to evaluate Sally’s conditional claim—that if Barb drinks the herbal tea daily for a week, then her pain will go away—for truth/falsity. To do so, we will consider various scenarios, the details of which will bear on that evaluation.

**Scenario #1**: Barb does in fact drink the tea every day for a week as prescribed, and, after doing so, lo and behold, her pain is gone. Sally was right! Has she proved that the tea works?

**Scenario #2**: Barb does as Sally said and drinks the tea every day for a week, but, after the week is finished, the pain remains, the same as ever. In this scenario, we would say that Sally was wrong: her conditional advice was false.

**Scenario #3**: Barb doesn’t drink the tea for a week; the antecedent is false. But in this scenario, it turns out that after the week is up, Barb’s pain has gone away; the consequent is true. What do we say about Sally’s advice—if you drink the tea, the pain will go away—in this set of circumstances?

**Scenario #4**: Again Barb does not drink the tea (false antecedent), and after the week is up, the pain remains (false consequent). What do we say about the Sally’s conditional advice in this scenario?

Perhaps you can see what I’m doing here. Each of the scenarios represents one of the rows in the truth-table definition for the horseshoe. Sally’s conditional claim has an antecedent—Barb drinks the tea every day for a week—and a consequent—Barb’s pain goes away. These are p and q, respectively, in the conditional. Each scenario corresponds to a line in the truth table for conditional, below.

The difficulty with conditional statements in a truth-functional language, is that we have to assign either true or false to each statement, and the only clear case we have here is Scenario #2. When the antecedent is true (Barb drinks the tea) but the consequent never happens (her pain did not go away), that clearly breaks the conditional, and proves that Sally was wrong.

The problem with the first scenario, where she drinks the tea and the pain goes away, is that causation is very hard to prove (another chapter in this book goes over causation in detail). It is possible that the pain went away for an entirely different reason and the tea had no causal effect. The problem with the third and forth scenarios, where Barb didn’t drink the tea, is that we haven’t even tested the tea, so how do we come to a conclusion about its connection to getting rid of pain?

With our truth-functional conditional, we go with an “innocent until proven guilty” approach. If the condition is met, but the consequent never happens, we’ve proved that the conditional was “guilty”—or false. We haven’t proved a causal connection in any of the other lines—in the first one, one test of the tea is not proof, and in the third and fourth lines we didn’t even test the tea—but we haven’t *disproved* it either, so we call it “true” in all of these cases. The table for the conditional looks like this:

|  |  |  |
| --- | --- | --- |
| p | q | p ⊃ q |
| T | T | T |
| T | F | F |
| F | T | T |
| F | F | T |

This works rather well with conditionals where we already know their truth value. Consider this one: “If I live in Macon, then I live in Georgia.” If a person lived in Macon but did not live in Georgia, then this would clearly be false—someone got their geography wrong. There are four cities named Macon in the U.S., so it is possible to live in a Macon without living in Georgia. I am talking about the one in Georgia, though, so if you do live there (true antecedent), then you do live in Georgia (true consequent). Consider, though, a person who does not live in Macon (false antecedent). They could live elsewhere in Georgia (true consequent), or they could live in, say, Tennessee (false consequent). The false antecedent does not tell us anything about the consequent—but someone not living in Macon doesn’t suddenly mean Macon is not a town in Georgia. The conditional remains true in these cases.

Another example: consider a promise. “If you give me gas money, then I’ll give you a ride to school.” Now consider the four scenarios, the four lines of the truth table, and consider when you have broken your promise and when you have not.

**Scenario #1**: Your friend gives you gas money, and you give them a ride. You did not break your promise, this was a true promise.

**Scenario #2**: Your friend gives you gas money; you take their money and run. You broke your promise—this was a false promise.

**Scenario #3**: Your friend tells you they’re completely broke (false antecedent), and they beg you to drive them to school anyway. You decide to be nice and drive them anyway (true consequent). This does not mean the promise was false, and you definitely haven’t broken your promise. The promise is still true, even though they couldn’t come up with the money this week.

**Scenario #4**: Your friend tells you they’re broke (false antecedent) and ask for a ride anyway; you decide you don’t feel like it (false consequent). You haven’t broken your promise here, either. The promise was still true, but they didn’t meet the condition, so you don’t have to give them a ride.

The point is, the conditional only kicks in when the antecedent is met. When they give you money, you have to give them a ride, or you’ve broken your promise. When they don’t give you money, you have no obligation either way, and you can make a choice without breaking anything. You told them what you’d do *if* they gave you gas money; you did not tell them what you’d do if they did *not* give you gas money.

The take-away: a conditional is only false when the antecedent is met but the consequent never happened – when the antecedent is true, but the consequent is false.

### 5. Biconditionals (TRIPLE-BAR)

As we said, biconditionals are, roughly, ‘if and only if’ sentences—sentences like “Beyoncé is logical if and only if James Brown is alive.” They’re called biconditionals because they contain two conditionals in them. Remember two of the phrases we symbolize with the triple bar: ‘if and only if’ and ‘necessary and sufficient condition.’ We can treat these phrases as a conjunction of two conditionals. So, “A if and only if B” can be broken down into “A if B, and A only if B.”

If you remember from the syntax section above, when you have an ‘if’ in the middle, as in “A if B,” you have to switch the order of the two sides. When you have an ‘only if’ in the middle, you need to keep them in the same order. So we can translate it like this:

parenthesis B horseshoe A end parenthesis, dot parenthesis A horseshoe B end parenthesis.Two conditionals, going in either direction. The triple bar is essentially short for this.

So, how do we figure out the truth conditions of this one? I think it helps if we stick with our “promise” example from the conditional section above. Suppose I tell my friend “I will give you a ride *if and only if* you pay me gas money.” A logically equivalent[[5]](#footnote-5) way to say the same thing is this: “If you pay me gas money, then I will give you a ride; and if you don’t pay me gas money, I will not give you a ride.” In symbols this would be:

parenthesis B horseshoe A end parenthesis, dot parenthesis tilde A horseshoe tilde B end parenthesis. Now you have told your friend what you would do if they give you that money, but you’ve also told them what you’ll do if they fail to pay you. If they don’t give you money, you no longer have a choice, you have to refuse the ride.

Take a look at the triple-bar truth table:

|  |  |  |
| --- | --- | --- |
| p | q | p ≡ q |
| T | T | T |
| T | F | F |
| F | T | F |
| F | F | T |

“If and only if”—two conditionals going in each direction. “Necessary and sufficient condition”—two conditionals, going in each direction. Remember the third phrase we use the triple-bar for: “logically equivalent to.” Notice that the triple-bar is true where both sides are true, and where both sides are false. It’s true when the two sides match. If there is a mismatch, either FT or TF, the biconditional is false.

## IV. Computing Truth-Values of Compound PL Sentences

With the truth-functional definitions of the five PL operators in hand, we can compute the truth-values of compound PL sentences, given the truth-values of their simplest parts (the simple sentences—capital letters). To do so, we must first determine what type of compound sentence we’re dealing with: negation, conjunction, disjunction, conditional, or biconditional. This involves deciding which of the operators in the PL sentence is the main operator. We then compute the truth-value of the compound by taking the values of the simpler components, and using the truth table for the operator to figure out the value of the whole sentence. If these components are themselves compound, we need to first determine their main operators and compute accordingly, in terms of their simpler components, and so on. A few examples will make the process clear.

Let’s suppose that A and B are true sentences. Consider this compound:

A wedge B

What is its truth value? This is pretty easy to figure out intuitively—it’s a disjunction, and I know that means at least one disjunct has to be true to make the whole sentence true. Both disjuncts are true, so this is a true sentence. Let’s walk through the steps of using our truth tables, though.

You always want to start by filling in the values of the simple sentences. Write them directly under each letter. We’ve been told that both A and B are true, so we write that:

The top row is A wedge B. The second row has a T under each letter.

Now we take the values of each side, and plug them into our disjunction table. (If you don’t have all five tables written out on a piece of paper you can reference easily, now is a good time to do so). Both sides are Ts; this puts us on the top row of our truth table, and we can see the answer is T—the whole sentences is true. Fill that in, and you have your answer:

The first row is A wedge B. The second row has a T under each letter, and also under the wedge.

Let’s look at a slightly more complicated one:

tilde A wedge tilde B

Now we have three connectives. We need to figure out the main connective before we begin. The main connective is always the last thing you compute, and that will give you the answer for the whole sentence. This is still a disjunction; the wedge is the main connective. So, before I figure out the value of the wedge, I need to first figure out the value of each side. I start the same way as before; I put the values of the simple sentences directly under each letter. Both A and B are true, so I fill that in:

The top row is tilde A wedge tilde B. The second row has a T under each letter.

I’m not done calculating each side, yet. Remember that a negation switches the value of whatever it attaches to. A and B are true, but NOT A and NOT B are both false. Fill that in:

The top line is tilde A wedge tilde B. The second line has a T under each letter, and an F under each tilde.

All that is left is to calculate the main connective. I will still use my wedge table, but I want to make sure the values I plug into it are the main connectives of each side. In this case, the tildes have the values I want to use. So, both not A and not B are false; this puts me on the fourth line of the wedge truth table, and that tells me the whole sentence is false. Fill that in under the wedge, and you have your answer:

The top line is tilde A wedge tilde B. The second line has a T under each letter, an F under each tilde, and an F under the wedge. The F under the wedge is red.

Once these compound sentences get complicated, you’ll have a long string of Ts and Fs. Use any tools you can think of to help you navigate. Different colors, circling things, whatever helps you focus on the correct T or F at the correct step.

Let’s walk through a similar one, only instead of a disjunction, this sentence is a negation:

tilde parenthesis A wedge B end parenthesis

In the previous example, the main connective was the wedge. The tildes attached directly to the letters, and the wedge brought the whole sentence together. Now, I’ve put the tilde in front of parentheses. That means it negates the whole disjunction instead of each individual letter, so it is now the main connective. I need to solve the problem inside parentheses before I apply the negation. Start the same way as before: put the T under each letter, and then use the wedge table to figure out the value of the disjunction. Since both sides are true, the disjunction will be true:

The first line is tilde parenthesis A wedge B end parenthesis. The second line has a T under each letter and a T under the wedge. The T under the wedge is red.

I’ve solved the problem inside parentheses: the disjunction is true. Now I apply the negation. I know that the tilde switches the value of whatever it attaches to. It attaches to a disjunction, and the disjunction is true, so the negation turns that to false, and that’s our final answer:

The first line is tilde parenthesis A wedge B end parenthesis. The second line has a T under each letter, and the wedge. It now also has an F under the tilde.

It will perhaps be useful to look at one more example, this time of a more complex PL sentence. Suppose again that A and B are true simple sentences, and that X and Y are false. Let’s compute the truth-value of the following compound sentence:

tilde parenthesis A dot X end parenthesis, horseshoe, parenthesis B wedge tilde Y end parenthesis.

As a first step, it’s useful to mark the truth-values of the simple sentences:

The first line is: tilde parenthesis A dot X end parenthesis, horseshoe, parenthesis B wedge tilde Y end parenthesis. The second line has a T under the A, an F under the X, a T under the B, and an F under the Y.

Now, we need to figure out what kind of compound sentence this is; what is the main operator? This sentence is a conditional; the main operator is the horseshoe. The tilde at the far left negates the first half of the sentence only. We need to compute the truth-values of both the antecedent and consequent before we can figure out the value of the conditional. Let’s take the antecedent first. The tilde negates the conjunction, so before we can know what the tilde does, we need to know the truth-value of the conjunction inside the parentheses. Conjunctions are true just in case both conjuncts are true; in this case, A is true but X is false, so the conjunction is false. If you forget that, plug the T and F into the conjunction table; it will tell you the answer is false. Now apply the tilde: it switches the value of the conjunction. Since the conjunction is false, its negation must be true:

The first line is: tilde parenthesis A dot X end parenthesis, horseshoe, parenthesis B wedge tilde Y end parenthesis. The second line has a T under the A, an F under the X, a T under the B, and an F under the Y. It now also has an F under the dot between A and X, and a T under the tilde before the first parenthesis. The T under the tilde is red.

So, the antecedent of our conditional is true. Let’s look at the consequent. There is a tilde directly in front of a letter, so I deal with that first. Y is false, so not Y must be true. That means both disjuncts are true. When I plug two trues into the wedge table, the answer is true. Let’s fill that in:

This is the same as the last one, only now there is an F under the tilde in front of the Y, and a red T under the wedge.When I figure out the main connective—the horseshoe—I need to make sure I’m looking at the main connective of each side—the tilde on the first side, and the wedge in the second side. Both the antecedent and consequent of the conditional are true, and looking at the horseshoe table, that makes the whole conditional true:

This is the same as before, only now there is a purple T underneath the horseshoe.

One final note: sometimes you only need partial information to make a judgment about the truth-value of a compound sentence. Look again at the truth table definitions of the two-place operators:

This is a table with four lines. The header has p, q, p dot q, p wedge q, p triple-bar q, and p horseshoe q. The column under the P reads TTFF. The column under the q reads TFTF. The column under the dot reads TFFF. The column under the wedge reads TTTF. The column under the triple-bar reads TFFT. The column under the horseshoe reads TFTT.

For three of these operators—the dot, wedge, and horseshoe—one of the rows is not like the others. For the dot: it only comes out true when both p and q are true, in the top row. For the wedge: it only comes out false when both p and q are false, in the bottom row. For the horseshoe: it only comes out false when p is true and q is false, in the second row.

Noticing this allows us, in some cases, to compute truth-values of compounds without knowing the truth-values of both sides. Suppose again that A is true and X is false, and let Q be a simple sentence with an unknown truth-value (it has one, like all of them must; I’m just not telling you what it is). Consider this compound:

A wedge Q

We know one of the disjuncts is true; we don’t know the truth-value of the other one. But we don’t need to! A disjunction is only false when both of its disjuncts are false; it’s true when even one of its disjuncts is true. A being true is enough to tell us the disjunction is true; the value of Q doesn’t matter.

Consider the conjunction:

X dot Q

We only know the truth-value of one of the conjuncts: X is false. That’s all we need to know to compute the truth-value of the conjunction. Conjunctions are only true when both of their conjuncts are true; they’re false when even one of them is false. X being false is enough to tell us that this conjunction is false.

Finally, consider these conditionals:

Q horseshoe A and X horseshoe QThey are both true. Conditionals are only false when the antecedent is true and the consequent is false; so they’re true whenever the consequent is true (as is the case in Q ⊃ A) and whenever the antecedent is false (as is the case in X ⊃ Q).

1. This chapter is based on *Fundamental Methods of Logic*, by Matthew Knachel. [↑](#footnote-ref-1)
2. I’m using the term “proposition” instead of “sentence” here because a single grammatical sentence can contain multiple propositions, and the same proposition can be communicated by a variety of different sentences. When I do use the term “sentence,” I’m usually using it as a synonym for “proposition,” rather than referring to the grammatical unit that spans from the capital letter up front to the period which ends it. [↑](#footnote-ref-2)
3. For those more interested in why this is the case, here you go! Most of us want to view ‘unless’ as a conditional—an if … then statement. Conditionals get their own symbol as we’ll see in a minute. It’s really a conditional and a negation. If I want to say “Amir will go to the movies, unless Jodie goes,” what this means is “If Jodie does not go to the movies, then Amir will.” It’s easy to remember that ‘unless’ is an if … then statement, and you add a ‘not’ to one side, but it’s hard to remember which side to add the ‘not’ to, and which side is the ‘if’ part of your conditional. It turns out that it’s logically equivalent to “Amir will go to the movies, or Jodie will go,” so we’ll translate ‘unless’ using a wedge. [↑](#footnote-ref-3)
4. Ok, to be clear, Edison leaned heavily on the electrical inventions of scientists before him, and an English scientist named Joseph Swan independently invented a successful light bulb, around the same time Edison did. This happens in science more than you’d think it would. With inventions, the first person to get a patent gets credit. The light bulb case is complicated because Edison got his patent in the U.S., and Swan got one in the U.K. My fifteen minutes of googling turned up a wide variety of patent dates for both men. We still give the credit to Edison, so we’re going to call “Edison invented the light bulb” a true statement. [↑](#footnote-ref-4)
5. For those interested in how this is logically equivalent: I’ve repeated my second conjunct “if A then B.” The first conjunct is “if B then A,” which is logically equivalent to its contraposition: “if not A then not B.”” [↑](#footnote-ref-5)