

# CHAPTER 4

## Deductive Logic II: Sentential Logic

### I. Why Another Deductive Logic?

Aristotle's logic was great. It had a two-plus millennium run as the only game in town. As recently as the late 18<sup>th</sup> century (remember, Aristotle did his work in the 4<sup>th</sup> century BCE), the great German philosopher Immanuel Kant remarked that “since the time of Aristotle [logic] has not had to go a single step backwards... [and] it has also been unable to take a single step forward, and therefore seems to all appearance to be finished and complete.”<sup>1</sup>

That may have been the appearance in Kant's time, but only because of an accident of history. In his own time, in ancient Greece, Aristotle's system had a rival—the logic of the Stoic school, culminating in the work of Chrysippus. Recall, for Aristotle, the fundamental logical unit was the class; and since terms pick out classes, his logic is often referred to as a “term logic”. For the Stoics, the fundamental logical unit was the proposition; since sentences pick out propositions, we could call this a “sentential logic”. These two approaches to logic were developed independently. Because of the vicissitudes of intellectual history (more later commentators promoted Aristotelian Logic, original writings from Chrysippus didn't survive, etc.), it turned out that Aristotle's approach was the one passed on to future generations, while the Stoic approach lay dormant. However, in the 19<sup>th</sup> century, thanks to work by logicians like George Boole (and many others), the propositional approach was revived and developed into a formal system.

---

<sup>1</sup> Kant, I. 1997. *Critique of Pure Reason*. Guyer, P. and Wood, A. (tr.). Cambridge: Cambridge University Press. p. 106.

Why is this alternative approach valuable? One of the concerns we had when we were introducing Aristotelian Logic was that, because of the restriction to categorical propositions, we would be limited in the number and variety of actual arguments we could evaluate. We brushed aside these concerns with a (somewhat vague) promise that, as a matter of fact, lots of sentences that were not standard form categoricals could be translated into that form. Furthermore, the restriction to categorical syllogisms was similarly unproblematic (we assured ourselves), because lots of arguments that are not standard form syllogisms could be rendered as (possibly a series of) such arguments.

These assurances are true in a large number of cases. But there are some very simple arguments that resist translation into strict Aristotelian form, and for which we would like to have a simple method for judging them valid. Here is one example:

Either Clinton will win the election or Trump will win the election.

Trump will not win the election.

∴ Clinton will win the election.

None of the sentences in this argument is in standard form. And while the argument has two premises and a conclusion, it is not a categorical syllogism. Could we translate it into that form? Well, we can make some progress on the second premise and the conclusion, noting, as we did in Chapter 3, that there's a simple trick for transforming sentences with singular terms (names like 'Clinton' and 'Trump') into categoricals: let those names be class terms referring to the unit class containing the person they refer to, then render the sentences as universal. So the conclusion, 'Clinton will win the election' can be rewritten in standard form as 'All Clintons are election-winners', where 'Clintons' refers to the unit class containing only Hillary Clinton. Similarly, 'Trump will not win the election' could be rewritten as a universal negative: 'No Trumps are election-winners'. The first premise, however, presents some difficulty: how do I render an either/or claim as a categorical? What are my two classes? Well, election-winners is still in the mix, apparently. But what to do with Clinton and Trump? Here's an idea: stick them together into the same class (they're not gonna like this), a class containing just the two of them. Let's call the class 'candidates'. Then this universal affirmative plausibly captures the meaning of the original premise: 'All election-winners are candidates'. So now we have this:

All election-winners are candidates.

No Trumps are election-winners.

∴ All Clintons are election-winners.

At least all the propositions are now categoricals. The problem is, this is not a categorical syllogism. Those are supposed to involve exactly three classes; this argument has four—Clintons, Trumps, election-winners, and candidates. True, candidates is just a composite class made by combining Clintons and Trumps, so you can make a case that there are really only three classes here. But, in a categorical syllogism, each of the class terms is supposed to occur exactly twice. 'Election-winners' occurs in all three, and I don't see how I can eliminate one of those occurrences.

Ugh. This is giving me a headache. It shouldn't be this hard to analyze this argument. You don't have to be a logician (or a logic student who's made it through three chapters of this book) to

recognize that the Trump/Clinton argument is a valid one. Pick a random person off the street, show them that argument, and ask them if it's any good. They'll say it is. It's easy for regular people to make such a judgment; shouldn't it be easy for a logic to make that judgment, too? Aristotle's logic doesn't seem to be up to the task. We need an alternative approach.

This particular example is exactly the kind of argument that begs for a proposition-focused logic, as opposed to a class-focused logic like Aristotle's. If we take whole propositions as our fundamental logical unit, we can see that the form of this argument—the thing, remember, that determines its validity—is something like this:

Either p or q  
Not q  
 $\therefore$  p

In this schema, 'p' stands for the proposition that Clinton will win and 'q' for the proposition that Trump will win. It's easy to see that this is a valid form.<sup>2</sup> This is the advantage of switching to a sentential, rather than a term, logic. It makes it easy to analyze this and many other argument forms.

In this chapter, we will discuss the basics of the proposition-centered approach to deductive logic—Sentential Logic. As was the case with Aristotle's logic, Sentential Logic must accomplish three tasks:

1. Tame natural language.
2. Precisely define logical form.
3. Develop a way to test logical forms for validity.

The approach to the first task—taming natural language—will differ substantially from Aristotle's. Whereas Aristotelian Logic worked within a well-behaved portion of natural language—the sentences expressing categorical propositions—Sentential Logic steps outside of natural language entirely, constructing an artificial language and only evaluating arguments expressed in its terms. This move, of course, raises the concern we had about the applicability to everyday arguments even more acutely: what good is a logic if it doesn't evaluate English arguments at all? What we must show to alleviate this concern is that there is a systematic relationship between our artificial language and our natural one (English); we must show how to translate between the two—and how translating *from* English *into* the artificial language results in the removal of imprecision and unruliness, the taming of natural language.

We will call our artificial language "SL," short for 'Sentential Logic'. In constructing a language, we must specify its *syntax* and its *semantics*. The syntax of a language is the rules governing what

---

<sup>2</sup> This form is often called the "Disjunctive Syllogism". Notice that the word 'syllogism' is used there. By the Middle Ages, Stoic Logic hadn't disappeared entirely. Rather, bits of it were simply added on the Aristotelian system. So, it was traditional (and still is in many logic textbooks), when discussing Aristotelian Logic, to present this form, along with some others, as additional valid forms (supplementing Barbara, Datisi, and the rest). But this conflation of the two traditions obscures the fundamental difference between a class-centered approach to logic and one focused on propositions. These should be kept distinct.

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. The semantics of a language is an account of the meanings of its well-formed bits. If you know what a sentence means, then you know what it takes for it to express a truth or a falsehood. So semantics tells you under what conditions a given proposition is true or false.<sup>3</sup> Our discussion of the semantics of SL will reveal its relationship to English and tell us how to translate between the two languages.

## II. Syntax of SL

First, we cover syntax. This discussion will give us some clues as to the relationship between SL and English, but a full accounting of that relationship will have to wait, as we said, for the discussion of semantics.

We can distinguish, in English, between two types of (declarative) sentences: *simple* and *compound*. A simple sentence is one that does not contain any other sentence as a component part. A compound sentence is one that contains at least one other sentence as a component part. (We will not give a rigorous definition of what it is for one sentence to be a *component part* of another sentence. Rather, we will try to establish an intuitive grasp of the relation by giving examples, and stipulate that a rigorous definition *could* be provided, but is too much trouble to bother with.) 'Beyoncé is logical' is a simple sentence; none of its parts is itself a sentence.<sup>4</sup> 'Beyoncé is logical and James Brown is alive' is a compound sentence: it contains two simple sentences as component parts—namely, 'Beyoncé is logical' and 'James Brown is alive'.

In SL, we will use capital letters—'A', 'B', 'C', ..., 'Z'—to stand for simple sentences. Our practice will be simply to choose capital letters for simple sentences that are easy to remember. For example, we can choose 'B' to stand for 'Beyoncé is logical' and 'J' to stand for 'James Brown is alive'. Easy enough. The hard part is symbolizing compound sentences in SL. How would we handle 'Beyoncé is logical and James Brown is alive', for example? Well, we've got capital letters to stand for the simple parts of the sentence, but that leaves out the word 'and'. We need more symbols.

We will distinguish five different kinds of compound sentence, and introduce a special SL symbol for each. Again, at this stage we are only discussing the syntax of SL—the rules for combining its symbols into well-formed constructions. We will have some hints about the semantics of these

---

<sup>3</sup> That's actually a controversial claim about the role of semantics. Your humble author, for example, is one of the weirdos who thinks it not true (of natural language, at least). But let's leave those deviant linguists and philosophers (and their abstruse arguments) to one side and just say: semantics gives you truth-conditions. That's certainly true of our artificial language SL.

<sup>4</sup> You might think 'Beyoncé is' is a part of the sentence that qualifies as a sentence itself—a sentence claiming that she exists, maybe. But that won't do. The word 'is' in the original sentence is the "'is' of predication"—a mere linking verb; 'Beyoncé is' only counts as a sentence if you change the meaning of 'is' to the "'is' of existence". Anyway, stop causing trouble. This is why we didn't give a rigorous definition of 'component part'; we'd get bogged down in these sorts of arcane distinctions.

new symbols—hints about their meanings—but a full treatment of that topic will not come until the next section.

## Conjunctions

The first type of compound sentence is one that we’ve already seen. Conjunctions are, roughly, ‘and’-sentences—sentences like ‘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 SL, that symbol is a “dot”. It looks like this: •.

To form a conjunction in SL, we simply stick the dot between the two component letters, thus:

$$B \bullet J$$

That is the SL version of ‘Beyoncé is logical and James Brown is alive’.

A note on terminology. 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 “left-hand conjunct” (‘B’ in this case) or the “right-hand conjunct” (‘J’ in this case).

## Disjunctions

Disjunctions are, roughly, ‘or’-sentences—sentences like ‘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 SL, that symbol is a “wedge”. It looks like this:  $\vee$ .

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

$$B \vee J$$

That is the SL version of ‘Beyoncé is logical or James Brown is alive’.

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 “left-hand disjunct” (‘B’ in this case) or the “right-hand disjunct” (‘J’ in this case).

## Negations

Negations are, roughly, ‘not’-sentences—sentences like ‘James Brown is not alive’. You may find it surprising that this would be considered a compound sentence. It is not immediately clear how any component part of this sentence is itself a sentence. Indeed, if the definition of ‘component part’ (which we intentionally have not provided) demanded that parts of sentences contain only contiguous words (words next to each other), you couldn’t come up with a part of ‘James Brown is not alive’ that is itself a sentence. But that is not a condition on ‘component part’. In fact, this

sentence does contain another sentence as a component part—namely, ‘James Brown is alive’. This can be made more clear if we paraphrase the original sentence. ‘James Brown is not alive’ means the same thing as ‘It is not the case that James Brown is alive’. Now we have all the words in ‘James Brown is alive’ next to each other; it is clearly a component part of the larger, compound sentence. We have ‘J’ to stand for the simple component; we need a symbol for ‘it is not the case that’. In SL, that symbol is a “tilde”. It looks like this:  $\sim$ .

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

$$\sim J$$

This is the SL version of ‘James Brown is not alive’.

## Conditionals

Conditionals are, roughly, ‘if/then’ sentences—sentences like ‘If Beyoncé is logical, then James Brown is alive’. (James Brown is actually dead. But suppose Beyoncé is a “James Brown-truther”, a thing that I just made up. She claims that James Brown faked his death, that the Godfather of Soul is still alive, getting funky in some secret location.<sup>5</sup> In that case, the conditional sentence might make sense.) 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 SL, that symbol is a “horseshoe”. It looks like this:  $\supset$ .

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

$$B \supset J$$

That is the SL version of ‘If Beyoncé is logical, then James Brown is alive’.

A note on terminology. Unlike our treatment of conjunctions and disjunctions, we will distinguish between the two components of the conditional. The component to the left of the horseshoe will be called the “antecedent” of the conditional; the component after the horseshoe is its “consequent”. As we will see when we get to the semantics for SL, there is a good reason for distinguishing the two components.

## Biconditionals

Biconditionals are, roughly, ‘if and only if’-sentences—sentences like ‘Beyoncé is logical if and only if James Brown is alive’. (This is perhaps not a familiar locution. We will talk more about what it means when we discuss semantics.) 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 SL, that symbol is a “triple-bar”. It looks like this:  $\equiv$ .

---

<sup>5</sup> Play along.

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

$$B \equiv J$$

That is the SL version of ‘Beyoncé is logical if and only if James Brown is alive’.

There are no special names for the components of the biconditional.

## Punctuation – Parentheses

Our language, SL, 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 two more: the left- and right-hand parentheses. And that’ll be it.

We use parentheses in SL for one reason (and one reason only): to remove ambiguity. To see how this works, it will be helpful to draw an analogy between SL 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:

$$2 + 3 \times 5$$

As it stands, this formula is ambiguous. 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.<sup>6</sup> We can use parentheses to disambiguate, and we can do so in two different ways:

$$(2 + 3) \times 5$$

or

$$2 + (3 \times 5)$$

And 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. And it comes out to 17. Different placement of parentheses, different results.

This same sort of thing is going to arise in SL. We use the same term we use to refer to the addition and multiplication signs—‘operator’—to refer to dot, wedge, tilde, horseshoe, and triple-bar. (As we will see when we look at the semantics for SL, this is entirely proper, since the SL operators

---

<sup>6</sup> You may have learned an “order of operations” in grade school, according to which multiplication takes precedence over addition, so that there would be no ambiguity in this expression. But the order of operations is just a (mostly arbitrary) way of removing ambiguity that would be there without it. The point is, absent some sort of disambiguating convention—whether it’s parentheses or an order of operations—the meanings of expressions like this are indeterminate.

stand for mathematical functions on truth-values.) There are ways of combining SL 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 sentence, but it contains both the word 'and' and the 'if/then' construction. And 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 SL symbols work, we would render the sentence like this:

$$B \bullet J \supset Q$$

But just as was the case with the arithmetical example above, this formula is ambiguous. I don't know what kind of compound sentence 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:

$$(B \bullet J) \supset Q$$

or

$$B \bullet (J \supset Q)$$

The first formula is a conditional; horseshoe is its main operator, and its antecedent is a compound sentence (the conjunction 'B • J'). The second formula is a conjunction; dot is its main operator, and its right-hand conjunct is a compound sentence (the conditional 'J ⊃ Q'). 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'.

The question is, what kind of compound sentence is the original? Is it a conditional or a conjunction? It is not a conjunction. Conjunctions are, roughly (again, we're not really doing semantics yet), 'and'-sentences. When you utter a conjunction, you're committing yourself to both of the conjuncts. If I say, "Beyoncé is logical and James Brown is alive," I'm telling you that both of those things are true. If we construe the present sentence as a conjunction, properly symbolized as 'B • (J ⊃ Q)', then we take it that the person uttering the sentence is committed to both conjuncts; she's telling us that two things are true: (1) Beyoncé is logical *and* (2) if James Brown is alive then she's the Queen of England. So, if we take this to be a conjunction, we're interpreting the speaker as committed to the proposition that Beyoncé is logical. But clearly she's not. She uttered 'If Beyoncé is logical and James Brown is alive, then I'm the Queen of England' to express dubiousness about Beyoncé's logicity (and James Brown's status among the living). This sentence is not a conjunction; it is a conditional. It's saying that *if* those two things are true (about Beyoncé and James Brown), then I'm the Queen of England. The utterer doubts both conjuncts in the antecedent. The proper symbolization of this sentence is the first one above: '(B • J) ⊃ Q'.

Again, in SL, 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.

Notice that I didn't mention the tilde there. Tilde is different from the other four. Dot, wedge, horseshoe, and triple-bar are what we might call "two-place operators". There are two simpler components in conjunctions, disjunctions, conditionals, and biconditionals. Negations, on the other hand, have only one simpler component; hence, we might call tilde a "one-place operator". It only operates on one thing: the sentence it negates.

This distinction is relevant to our discussion of parentheses and ambiguity. We will adopt a convention according to which the tilde negates the first well-formed SL construction immediately to its right. This convention will have the effect of removing potential ambiguity without the need for parentheses. Consider the following combination of SL symbols:

$$\sim A \vee B$$

It may appear that this formula is ambiguous, with the following two possible ways of disambiguating:

$$\sim (A \vee B)$$

or

$$(\sim A) \vee B$$

But this is not the case. Given our convention—tilde negates the first well-formed SL construction immediately to its right—the original formula—' $\sim A \vee B$ '—is *not* ambiguous; it is well-formed. Since 'A' is itself a well-formed SL construction (of the simplest kind), the tilde in ' $\sim A \vee B$ ' negates the 'A' only. This means that we don't have to indicate this fact with parentheses, as in the second of the two potential disambiguations above. That kind of formula, with parentheses around a tilde and the item it negates, is not a well-formed construction in SL. Given our convention about tildes, the parentheses around ' $\sim A$ ' are redundant.

The first potential disambiguation—' $\sim (A \vee B)$ '—*is* well-formed, and it means something different from ' $\sim A \vee B$ '. In the former, the tilde negates the entire disjunction, ' $A \vee B$ '; in the latter, it only negates 'A'. That makes a difference. Again, an analogy to arithmetic is helpful here. Compare the following two formulas:

$$-(2 + 5)$$

vs.

$$-2 + 5$$

In the first, the minus-sign covers the entire sum, and so the result is -7; in the second, it only covers the 2, so the result is 3. This is exactly analogous to the difference between ' $\sim (A \vee B)$ ' and ' $\sim A \vee B$ '. The tilde has wider scope in the first formula, and that makes a difference. The difference can only be explained in terms of meaning—which means it is time to turn our attention to the semantics of SL.

### III. Semantics of SL

Our task is to give precise meanings to all of the well-formed formulas of SL. We will refer to these, quite sensibly, as “sentences of SL”. 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. While the semantics for a natural language like English is complicated (What is the meaning of a sentence? Its truth-conditions? The proposition expressed? Are those two things the same? Is it something else entirely? Ugh.), the semantics for SL sentences is simple: all we care about is truth-value. A sentence in SL can have one of two semantic values: true or false. That's it.

This is one of the ways in which the move to SL is a taming of natural language. In SL, 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 non-declarative sentences, which don't express propositions and don't have truth-values at all.<sup>7</sup> But even if we restrict ourselves to declarative English sentences, things don't look quite as simple as they are in SL. 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 18<sup>th</sup>, early 19<sup>th</sup> centuries); people were shorter back then (nutrition wasn't what it is these days, e.g.), and so Napoleon was about average or slightly above. People are taller now, though, so 5' 7" might be considered short. At least, short for a man. A grown man, that is. I mean, a grown man who's not a dwarf. Er, also, a grown non-dwarf man of French extraction (he'd be a tall man in Cambodia, for example, where the average height is only 5' 4"). The average height for a modern Frenchman is 5' 9.25". Napoleon is 2.25 inches shorter than average. Is that 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 is short'. So, in English, we might say that they lack a truth-value (absent some explicit specification of the relevant standards). Logics that are more sophisticated than our SL have developed ways to deal with these sorts of cases. Instead of just two truth-values, some logics add more. There are three-values logics, where you have true, false, and neither. So we could say 'Napoleon is short' is neither. There are logics 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); in such a system, we could assign, I don't know, .62 to the proposition that Napoleon is short. The point is, English and other natural languages are messy when it comes to truth-value. We're taming them in SL by assuming that every SL sentence

---

<sup>7</sup> Pausing briefly to note, once again, that this talk of *sentences*, rather than the propositions that they express, having truth-values is a bit fast and loose. Reaffirming our earlier stance on this: not a big deal.

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 from here is to provide semantics for the five operators: tilde, dot, wedge, triple-bar, and horseshoe (we start with tilde because it’s the simplest, and we save horseshoe for last because it’s quite a bit more involved). 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 sentence) resulting from the possible truth-value inputs (of the simpler components).

## Negations (TILDE)

Because tilde is a one-place operator, this is the simplest operator to deal with. The general form of a negation is  $\sim p$ , where ‘p’ is a variable standing for any generic SL sentence, simple or compound. As a lower-case letter, ‘p’ is not part of our language (SL); rather, it’s a tool we use to talk *about* our language—to refer to generic well-formed constructions within it.

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 SL 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. 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	$\sim p$
T	F
F	T

In the left-hand column, we have ‘p’, which, as a variable, stands for a generic, unspecified SL sentence. Since it’s unspecified, we don’t know its truth-value; but since it’s a sentence in SL, we do know that there are only two possibilities for its truth-value: true or false (T or F). So in the first column, we list those two possibilities. In the second column, we have ‘ $\sim p$ ’, the negation of whatever ‘p’ is. 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. This is what we represent when we write ‘F’ and ‘T’ underneath the tilde (the operator that effects the change in truth-value) in the second column, in the same rows as their opposites.

Tilde is a *truth-functional operator*. Its meaning is specified by a function: if you input a T, the output is an F; if you input an F, the output is a T. The other four operators will also be defined in

terms of the truth-function they represent. This is exactly analogous, again, to arithmetic. Addition, with its operator ‘+’, is a function on numbers. Input 1 and 3, and the output is 4. In SL, we only have two values—T and F—but it’s the same kind of thing. We could just as well use numbers to represent the truth-values: 0 for false and 1 for true, for example. In that case, tilde would be a function that outputs 0 when 1 is the input, and outputs 1 when 0 is the input.

## 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. 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 SL is  $p \bullet q$ . 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. If I tell you that Beyoncé is wise and James Brown is alive, I’m committing myself to the truth of both of those alleged facts; I am, as it were, promising you that both of those things are true. So, if even one of them turns out false, I’ve broken my promise; the only way the promise is kept is if both of them turn out to be true.

This is how conjunctions work, then: they’re true just in case both conjuncts are true; false otherwise. We can represent this graphically, with a truth-table defining the dot:

p	q	$p \bullet q$
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 SL sentence that can be either true or false. That gives us four possibilities for their truth-values as a pair: both true,  $p$  true and  $q$  false,  $p$  false and  $q$  true, 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.

## Disjunctions (WEDGE)

Our rough characterization of disjunctions was that they are ‘or’-sentences—sentences like ‘Beyoncé is logical or James Brown is alive’. In SL, the general form of a disjunction is  $p \vee q$ . We need to figure out the circumstances in which such a compound is true; we need the truth-function represented by the wedge.

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: ‘Clinton will win the election or Trump will win the election’. 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 the surf ‘n’ turf (lobster and steak combined in the same entrée), 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.<sup>8</sup> We need to choose a meaning for wedge, and we’re choosing the inclusive sense of ‘or’. 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 SL, using a combination of operators.

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

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

## 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’. ‘If and only if’ is not a phrase most people use in everyday life, but the meaning is straightforward: it’s used to claim that both components have the same truth-value, that one entails the other and *vice versa*, that they can’t have different truth-values. In SL, the general form of a biconditional is  $p \equiv q$ . This is the truth-function:

---

<sup>8</sup> As was the case when we had to make a choice about the word ‘some’ in Aristotelian logic, the argument makes the case that the inclusive sense is the core meaning of ‘or’, and the exclusive sense is a meaning that’s often, but not always, conveyed when we use ‘or’ in particular circumstances—an implicature. This line of reasoning has both adherents and detractors.

p	q	$p \equiv q$
T	T	T
T	F	F
F	T	F
F	F	T

The triple-bar is kind of like a logical equals-sign (it even resembles ‘=’): the function delivers an output of T when both components are the same, F when they’re not.

While the truth-functional meaning of triple-bar is now clear, it still may be the case that the intuitive meaning of the English phrase ‘if and only if’ remains elusive. This is natural. Fear not: we will have much more to say about that locution when we discuss translating between English and SL; a full understanding of *biconditionals* can only be achieved based on a full understanding of *conditionals*, to which, as the names suggest, they are closely related. We now turn to a specification of the truth-functional meaning of the latter.

## 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 SL. 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 SL is  $p \supset q$ . We need to decide for which values of p and q the conditional turns out true and false. To help us along (by making things more vivid), we’ll consider an actual conditional claim, with a little story to go along with it. Suppose Barb is suffering from joint pain; maybe it’s gout, maybe it’s arthritis—she doesn’t know and hasn’t been to the doctor to find out. She’s complaining about her pain to her neighbor, Sally. Sally is a big believer in “alternative medicine” and “holistic healing”. 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! In this scenario, we would say that the conditional we’re evaluating is true.

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.

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  $p \supset q$ . In scenario #1, both  $p$  and  $q$  were true: Barb did drink the tea, and the pain did go away; in scenario #2,  $p$  was true (Barb drank the tea) but  $q$  was false (the pain didn't go away). These two scenarios are the first two rows of the four-row truth tables we've already seen for dot, wedge, and triple-bar. For horseshoe, the truth-function gives us T in the first row and F in the second:

$p$	$q$	$p \supset q$
T	T	T
T	F	F

All that's left is to figure out what happens in the third and fourth rows of the table, where the antecedent ( $p$ , Barb drinks the tea) is false both times and the consequent is first true (in row 3) and then false (in row 4). There are two more scenarios to consider.

In scenario #3, Barb decides Sally is a bit of a nut, or she drinks the tea once and it tastes awful so she decides to stop drinking it—whatever the circumstances, 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?

In 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?

It's tempting to say that in the 3<sup>rd</sup> and 4<sup>th</sup> scenarios, since Barb didn't even *try* Sally's remedy, we're not in a position to evaluate Sally's advice for truth or falsity. The hypothesis wasn't even tested. So, we're inclined to say 'If you drink the tea, then the pain will go away' is neither true nor false. But while this might be a viable option in English, it won't work in SL. We've made the simplifying assumptions that every SL sentence must have a truth-value, and that that the only two possibilities are true and false. We can't say it has no truth-value; we can't add a third value and call it "neither". We have to put a T or an F under the horseshoe in the third and fourth rows of the truth table for that operator. Given this restriction, and given that we've already decided how the first two rows should work out, there are four possible ways of specifying the truth-function for horseshoe:

p	q	(1)	(2)	(3)	(4)
T	T	T	T	T	T
T	F	F	F	F	F
F	T	F	T	F	T
F	F	F	F	T	T

These are our only options (remember, the top two rows are settled; scenarios 1 and 2 above had clear results). Which one captures the meaning of the conditional best?

Option 1 is tempting: as we noted, in rows 3 and 4, Sally's hypothesis isn't even tested. If we're forced to choose between true and false, we might as well go with false. The problem with this option is that this truth-function—true when both components are true; false otherwise—is already taken. That's the meaning of dot. If we choose option 1, we make horseshoe and dot mean the same thing. That won't do: they're different operators; they should have different meanings. 'And' and 'if/then' don't mean the same thing in English, clearly.

Option 2 also has its charms. OK, we might say, in neither situation is Sally's hypothesis tested, but at least row 3 has something going for it, Sally-wise: the pain does go away. So let's say her conditional is true in that case, but false in row 4 when there still is pain. Again, this won't do. Compare the column under option 2 to the column under  $q$ . They're the same: T, F, T, F. That means the entire conditional,  $p \supset q$ , has the same meaning as its consequent, plain old  $q$ . Not good. The antecedent,  $p$ , makes no difference to the truth-value of the conditional in this case. But it should; we shouldn't be able to compute the truth-value of a two-place function without even looking at one of the inputs.

Option 3 is next. Some people find it reasonable to say that the conditional is false in row 3: there's something about the disappearance of the pain, despite not drinking the tea, that's incompatible with Sally's prediction. And if we can't put an F in the last row too (this is just option 1 again), then make it a T. But this fails for the same reason option 1 did: the truth-function is already taken, this time by the triple-bar. 'If and only if' is a much stronger claim than the mere 'if/then'; biconditionals must have a different meaning from mere conditionals.

That leaves option 4. This is the one we'll adopt, not least because it's the only possibility left. The conditional is true when both antecedent and consequent are true—scenario 1; it's false when the antecedent is true but the consequent false—scenario 2; and it's true whenever the antecedent is false—scenarios 3 and 4. This is the definition of horseshoe:

p	q	$p \supset q$
T	T	T
T	F	F
F	T	T
F	F	T

It's not ideal. The first two rows are quite plausible, but there's something profoundly weird about saying that the sentence 'If you drink the tea, then the pain will go away' is true whenever the tea is not drunk. Yet that is our only option. We can perhaps make it a bit more palatable by saying—as we did about universal categorical propositions with empty subject classes—that while it's true in such cases, it's only true vacuously or trivially—true in a way that doesn't tell you about how things are in the world.

What can also help a little is to point out that while rows 3 and 4 don't make much sense for the Barb/Sally case, they do work for other conditionals. The horror author Stephen King lives in Maine (half his books are set there, it seems). Consider this conditional: 'If Stephen King is the Governor of Maine, then he lives in Maine'. While a prominent citizen, King is not Maine's governor, so the antecedent is false. He is, though, as we've noted, a resident of Maine, so the consequent is true. We're in row 3 of the truth-table for conditionals here. And intuitively, the conditional is true: he's not the governor, but *if he were*, he would live in Maine (governors reside in their states' capitals). And consider this conditional: 'If Stephen King is president of the United States, then he lives in Washington, DC'. Now both the antecedent (King is president) and the consequent (he lives in DC) are false: we're in row 4 of the table. But yet again, the conditional claim is intuitively true: *if he were* president, he would live in DC.

Notice the trick I pulled there: I switched from the so-called indicative mood (if he *is*) to the subjunctive (if he *were*). The truth of the conditional is clearer in the latter mood than the former. But this trick won't always work to make the conditional come out true in the third and fourth rows. Consider: 'If Stephen King were president of the United States, then he would live in Maine' and 'If Stephen King were Governor of Maine, then he would live in Washington, DC'. These are third and fourth row examples, respectively, but neither is intuitively true.

By now perhaps you are getting a sense of why conditionals are such a vexed topic in the history of logic. A variety of approaches, with attendant alternative logical formalisms, have been developed over the centuries (and especially in the last century) to deal with the various problems that arise in connection with conditional claims. Ours is the very simplest approach, the one with which to begin. As this is an introductory text, this is appropriate. You can investigate alternative accounts of the conditional if you extend your study of logic further.

## Computing Truth-Values of Compound SL Sentences

With the truth-functional definitions of the five SL operators in hand, we can develop a preliminary skill that will be necessary to deploy when the time comes to test SL arguments for validity. We need to be able to compute the truth-values of compound SL 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 SL sentence is the *main* operator. We then compute the truth-value of the compound according to the definition for the appropriate operator, using the truth-values of the simpler components. If these components are themselves compound, we determine their main operators and compute accordingly, in terms of their simpler components—repeating as necessary until we get down to the simplest components of all, the capital letters. A few examples will make the process clear.

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

$$\sim A \vee B$$

What is its truth-value? To answer that question, we first have to figure out what kind of compound sentence we're dealing with. It has two operators—the tilde and the wedge. Which of these is the main operator; that is, do we have a negation or a disjunction? We answered this question earlier, when we were discussing the syntax of SL. Our convention with tildes is that they negate the first well-formed construction immediately to their right. In this case, 'A' is the first well-formed construction immediately to the right of the tilde, so the tilde negates it. That means wedge is the main operator; this is a disjunction, where the left-hand disjunct is  $\sim A$  and the right-hand disjunct is B. To compute the truth-value of the disjunction, we need to know the truth-values of its disjuncts. We know that B is true; we need to know the truth-value of  $\sim A$ . That's easy, since A is true,  $\sim A$  must be false. It's helpful to keep track of one's step-by-step computations like so:

$$\begin{array}{cc} T & T \\ \sim A & \vee B \\ F & \end{array}$$

I've marked the truth-values of the simplest components, A and B, on top of those letters. Then, under the tilde, the operator that makes it happen, I write 'F' to indicate that the left-hand disjunct,  $\sim A$ , is false. Now I can compute the truth-value of the disjunction: the left-hand disjunct is false, but the right hand disjunct is true; this is row 3 of the wedge truth-table, and the disjunction turns out true in that case. I indicate this with a 'T' under the wedge, which I highlight (with boldface and underlining) to emphasize the fact that this is the truth-value of the whole compound sentence:

$$\begin{array}{cc} T & T \\ \sim A & \vee B \\ F & \\ & \underline{\mathbf{T}} \end{array}$$

When we were discussing syntax, we claimed that adding parentheses to a compound like the last one would alter its meaning. We're now in a position to prove that claim. Consider this SL sentence (where A and B are again assumed to be true):

$$\begin{array}{cc} T & T \\ \sim (A \vee B) & \end{array}$$

Now the main operator is the tilde: it negates the entire disjunction inside the parentheses. To discover the effect of that negation on truth-value, we need to compute the truth-value of the disjunction that it negates. Both A and B are true; this is the top row of the wedge truth-table—disjunctions turn out true in such cases:

$$\begin{array}{cc} T & T \\ \sim (A \vee B) & \\ & T \end{array}$$

So the tilde is negating a truth, giving us a falsehood:

$$\begin{array}{cc} T & T \\ \sim (A \vee B) & \\ & T \\ \underline{F} & \end{array}$$

The truth-value of the whole is false; the similar-looking disjunction without the parentheses was true. These two SL sentences must have different meanings; they have different truth-values.

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

$$\sim (A \bullet X) \supset (B \vee \sim Y)$$

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

$$\begin{array}{cc} T & F & T & F \\ \sim (A \bullet X) \supset (B \vee \sim Y) & \end{array}$$

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 well-formed construction immediately to its right. In this case, that is  $(A \bullet X)$ .  $\sim (A \bullet X)$  is the antecedent of this conditional;  $(B \vee \sim Y)$  is the consequent. We need to compute the truth-values of each of these before we can compute the truth-value of the whole compound.

Let's take the antecedent,  $\sim (A \bullet X)$  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, and its negation must be true:

$$\begin{array}{cc} T & F & T & F \\ \sim (A \bullet X) \supset (B \vee \sim Y) & \\ & F & \\ T & \end{array}$$

So the antecedent of our conditional is true. Let's look at the consequent,  $(B \vee \sim Y)$ . Y is false, so  $\sim Y$  must be true. That means both disjuncts, B and  $\sim Y$  are true, making our disjunction true:

$$\begin{array}{cc} T & F & T & F \\ \sim (A \bullet X) \supset (B \vee \sim Y) & \\ & F & T & \\ T & & T & \end{array}$$

Both the antecedent and consequent of the conditional are true, so the whole conditional is true:

$$\begin{array}{cccc}
 & T & F & T & F \\
 \sim(A \bullet X) & \supset & (B \vee \sim Y) & & \\
 & F & & T & \\
 T & & & T & \\
 & \underline{T} & & &
 \end{array}$$

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:

p	q	$p \bullet q$	$p \vee q$	$p \equiv q$	$p \supset q$
T	T	T	T	T	T
T	F	F	T	F	F
F	T	F	T	F	T
F	F	F	F	T	T

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 components. Suppose again that A is true and X is false; and let Q be a simple SL sentence the truth-value of which is a mystery to you (it has one, like all of them must; I'm just not telling you what it is). Consider this compound:

$$A \vee Q$$

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; Q doesn't matter.

Consider the conjunction:

$$X \bullet Q$$

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:

$$Q \supset A \quad \text{and} \quad X \supset Q$$

They 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  $Q \supset A$ ) and whenever the antecedent is false (as is the case in  $X \supset Q$ ).

## EXERCISES

Compute the truth-values of the following compound sentences, where A, B, and C are true; X, Y, and Z are false; and P and Q are of unknown truth-value.

1.  $\sim B \vee X$
2.  $A \bullet \sim Z$
3.  $\sim X \supset \sim C$
4.  $(B \equiv C) \vee (X \bullet Y)$
5.  $\sim (C \vee (X \bullet \sim Y))$
6.  $(X \supset \sim A) \supset (\sim Z \bullet B)$
7.  $\sim (A \vee \sim X) \vee (C \bullet \sim Y)$
8.  $A \supset (\sim X \bullet \sim (C \equiv Y))$
9.  $\sim (\sim Z \supset \sim (\sim (A \equiv B) \vee \sim X))$
10.  $\sim (A \equiv \sim (\sim C \supset (B \bullet \sim X))) \equiv (\sim ((\sim Y \vee \sim A) \supset \sim B) \bullet (A \supset (((B \equiv \sim X) \equiv Z) \supset \sim Y)))$
11.  $\sim X \vee (Q \supset Z)$
12.  $\sim Q \bullet (A \supset (P \bullet \sim B))$
13.  $\sim (Q \bullet \sim Z) \supset (\sim P \supset C)$
14.  $\sim (\sim (A \bullet B) \supset ((P \vee \sim C) \equiv (\sim X \vee Q)))$
15.  $\sim P \vee (Q \vee P)$

## IV. Translating from English into SL

Soon we will learn how to evaluate arguments in SL—arguments whose premises and conclusions are SL sentences. In real life, though, we're not interested in evaluating arguments in some artificial language; we're interested in evaluating arguments presented in natural languages like English. So in order for our evaluative procedure of SL argument to have any real-world significance, we need to show how SL arguments can be fair representations of natural-language counterparts. We need to show how to translate sentences in English into SL.

We already have some hints about how this is done. We know that simple English sentences are represented as capital letters in SL. We know that our operators—tilde, dot, wedge, horseshoe, and triple-bar—are the SL counterparts of the English locutions 'not', 'and', 'or', 'if/then', and 'if and only if', respectively. But there is significantly more to say on the topic of the relationship between English and SL. Our operators—alone or in combination—can capture a much larger portion of English than that short list of words and phrases.

### Tilde, Dot, and Wedge

Consider the word 'but'. In English, it has a different meaning from the word 'and'. When I say "Donald Trump is rich and generous," I communicate one thing; when I say "Donald Trump is rich, but generous," I communicate something slightly different. Both utterances convey the assertions that Trump is rich, on the one hand, and generous on the other. The 'but'-sentence,

though, conveys something more—namely, that there’s something surprising about the generosity in light of the richness, that there’s some tension between the two. But notice that each of those utterances is true under the same circumstances: when Trump is both rich and generous; the difference between ‘but’ and ‘and’ doesn’t affect the truth-conditions. Since the meanings of our SL operators are specified entirely in terms of their effects on truth-values, SL is blind to the difference in meaning between ‘and’ and ‘but’. Since the truth-conditions for compounds featuring the two words are the same—true just in case both components are true, and false otherwise—we can use the dot to represent both. ‘Donald Trump is rich and generous’ and ‘Donald Trump is rich, but generous’ would both be rendered in SL as something like ‘ $R \bullet G$ ’ (where ‘ $R$ ’ stands for the simple sentence ‘Trump is rich’ and ‘ $G$ ’ stands for ‘Trump is generous’). Again, switching from English into SL is a strategy for dealing with the messiness of natural language: to conduct the kind of rigorous logical analyses involved in evaluating deductive arguments, we need a simpler, tamer language; the slight difference in meaning between ‘and’ and ‘but’ is one of the wrinkles we need to iron out before we can proceed.

There are other words and phrases that have the same effect on truth-value as ‘and’, and which can therefore be represented with the dot: ‘although’, ‘however’, ‘moreover’, ‘in addition’, and so on. These can all be used to form conjunctions.

There are fewer ways of forming disjunctions in English. Almost always, these feature the word ‘or’, sometimes accompanied by ‘either’. Whenever we see ‘or’, we will translate it into SL as the wedge. As we discussed, the wedge captures the inclusive sense of ‘or’—one or the other, or both. The exclusive sense—one or the other, but not both—can also be rendered in SL, using a combination of symbols. ‘Hillary Clinton or Donald Trump will win the election, but not both’. How would we translate that into SL? Let ‘ $H$ ’ stand for ‘Hillary Clinton will win’ and ‘ $D$ ’ stand for ‘Donald Trump will win’. We know how to deal with the ‘or’ part: ‘Hillary Clinton will win or Donald Trump will win’ is just ‘ $H \vee D$ ’. How about the ‘not both’ part? That’s the claim, paraphrasing slightly, that it’s not the case that both Hillary and Trump will win; that is, it’s the negation of the conjunction: ‘ $\sim (H \bullet D)$ ’. So we have the ‘or’ part, and we have the ‘not both’ part; the only thing left is the word ‘but’ in between. We just learned how to deal with that! ‘But’ gets translated as a dot. So the proper SL translation of ‘Hillary Clinton or Donald Trump will win the election, but not both’ is this:

$$(H \vee D) \bullet \sim (H \bullet D)$$

Notice we had to enclose the disjunction, ‘ $H \vee D$ ’, in parentheses. This is to remove ambiguity: without the parentheses, we wouldn’t know whether the wedge or the (middle) dot was the main operator, and so the construction would not have been well-formed. In SL, the exclusive sense of ‘or’ is expressed with a conjunction: it conjoins the (inclusive) ‘or’ claim to the ‘not both’ claim—one or the other, but not both.

It is worth pausing to reflect on the symbolization of ‘not both’, and comparing it to a complementary locution—‘neither/nor’. We symbolize ‘not both’ in SL as a negated conjunction; ‘neither/nor’ is a negated *disjunction*. The sentence ‘Neither Donald Trump nor Beyoncé will win the election’ would be rendered as ‘ $\sim (D \vee B)$ ’; that is, it’s not the case that either Donald or Beyoncé will win.

When we discussed the syntax of SL, it was useful to use an analogy to arithmetic to understand the interactions between tildes and parentheses. Taking that analogy too far in the case of negated conjunctions and disjunctions can lead us into error. The following is true in arithmetic:

$$-(2 + 5) = -2 + -5$$

We can distribute the minus-sign inside the parentheses (it's just multiplying by -1). The following, however, are *not* true in logic<sup>9</sup>:

$$\begin{array}{ll} \sim (p \bullet q) \equiv \sim p \bullet \sim q & [\text{WRONG}] \\ \sim (p \vee q) \equiv \sim p \vee \sim q & [\text{WRONG}] \end{array}$$

The tilde cannot be distributed inside the parentheses in these cases. For each, the left- and right-hand components have different meanings. To see why, we should think about some concrete examples. Let 'R' stand for 'Donald Trump is rich' and 'G' stand for 'Donald Trump is generous'. ' $\sim (R \bullet G)$ ' symbolizes the claim that Trump is not both rich and generous. Notice that this claim is compatible with his actually being rich, but not generous, and also with his being generous, but not rich. The claim is just that he's not *both*. Now consider the claim that ' $\sim R \bullet \sim G$ ' symbolizes. The main operator in that sentence is the dot; it's a conjunction. Conjunctions make a commitment to the truth of each of their conjuncts. The conjuncts in this case symbolize the sentences 'Trump is not rich' and 'Trump is not generous'. That is, this conjunction is committed to Trump's lacking both richness and generosity. That is a stronger claim than saying he's not both: if you say he's not both, that's compatible with him being one or the other; ' $\sim R \bullet \sim G$ ', on the other hand, insists that both are ruled out. So, generally speaking, a negated conjunction makes a different (weaker) claim than the conjunction of two negations.

There is also a difference between a negated disjunction and the disjunction of two negations. Consider ' $\sim (R \vee G)$ '. That symbolizes the sentence 'Trump is neither rich nor generous'. In other words, he lacks both richness and generosity. That's a much stronger claim than the one symbolized by ' $\sim R \vee \sim G$ '—the disjunction 'Either Trump isn't rich or he isn't generous'. He lacks one or the other quality (or both; the disjunction is inclusive). That's compatible with his actually being rich, but not generous; it's also compatible with his being generous, but not rich.

Did you notice what happened there? I used the same language to describe the claim symbolized by ' $\sim (R \bullet G)$ ' and ' $\sim R \vee \sim G$ '. Both merely assert that he isn't *both* rich and generous; he may be one or the other. I also described the claims made by ' $\sim (R \vee G)$ ' and ' $\sim R \bullet \sim G$ ' the same way. Both make the stronger claim that he lacks both characteristics. This is true in general: negated conjunctions are equivalent to the disjunction of two negations; and negated disjunctions are equivalent to the conjunction of two negations. The following logical equivalences are true<sup>10</sup>:

<sup>9</sup> The triple-bar is a logical equals-sign; it indicates that the components have the same truth-conditions (meaning).

<sup>10</sup> They're often referred to as "DeMorgan's Laws," after the nineteenth century English logician Augustus DeMorgan, who was apparently the first to formulate in the terms of the modern formal system developed by his fellow countryman and contemporary, George Boole. DeMorgan didn't discover these equivalences, however. They have been known to logicians since the ancient Greeks.

$$\sim (p \bullet q) \equiv \sim p \vee \sim q$$

$$\sim (p \vee q) \equiv \sim p \bullet \sim q$$

If you want to distribute that tilde inside the parentheses (or, alternatively, moving from right to left, pull the tilde outside), you have to change the wedge to a dot (and *vice versa*).

## Horseshoe and Triple-Bar

There are many English locutions that we can symbolize using the horseshoe and the triple-bar—especially the horseshoe. In fact, as we shall see, it's possible to render claims translated with the triple-bar using the horseshoe instead (along with a dot). We will look at a representative sample of the many ways in which conditionals and biconditionals can be expressed in English, and talk about how to translate them into SL using the horseshoe and triple-bar.

The canonical presentation of a conditional uses the words 'if' and 'then', as in 'If the Democrats win back Congress, then a lot of new legislation will be passed'. But the word 'then' isn't really necessary: 'If the Democrats win back Congress, a lot of new legislation will be passed' makes the same assertion. It would also be symbolized as ' $D \supset L$ ' (with 'D' and 'L' standing for the obvious simple components). The word 'if' can also be replaced. 'Provided the Democrats win back Congress, a lot of new legislation will be passed' also makes the same claim.

Things get tricky if we vary the placement of the 'if'. Putting it in the middle of sentence, we get 'Your pain will go away if you drink this herbal tea every day for a week', for example. Compare that sentence to the one we considered earlier: 'If you drink this herbal tea every day for a week, then your pain will go away'. Read one, then the other. They make the same claim, don't they? Rule of thumb: whatever follows the word 'if', when 'if' occurs on its own (without the word 'only'; see below), is the antecedent of the conditional. We would translate both of these sentences as something like ' $D \supset P$ ' (where 'D' is for drinking the tea, and 'P' is for the pain going away).

The word 'only' changes things. Consider: 'I will win the lottery only if I have a ticket'. A sensible claim, obviously true. I'm suggesting this is a conditional. Let 'W' stand for 'I win the lottery' and 'T' stand for 'I have a ticket'. Which is the antecedent and which is the consequent? Which of these two symbolizations is correct:

$$T \supset W$$

or

$$W \supset T$$

To figure it out, let's read them back into English as canonical 'if/then' claims. The first says, "If I have a ticket, then I'll win the lottery." Well, that's optimistic! But clearly false—something only a fool would believe. That can't be the correct way to symbolize our original, completely sensible claim that I will win only if I have a ticket. So it must be the second symbolization, which says that if I did win the lottery, then I had a ticket. That's better. Generally speaking, the component

occurring before ‘only if’ is the antecedent of a conditional, and the component occurring after is the consequent.

The claim in the last example can be put differently: having a ticket is a necessary condition for winning the lottery. We use the language of “necessary and sufficient conditions” all the time. We symbolize these locutions with the horseshoe. For example, being at least 16 years old is a necessary condition for having a driver’s license (in most states). Let ‘O’ stand for ‘I am at least 16 years old’ and ‘D’ stand for ‘I have a driver’s license.’  $D \supset O$  symbolizes the sentence claiming that O is necessary for D. The opposite won’t work:  $O \supset D$ , if we read it back, says “If I’m at least 16 years old, then I have a driver’s license.” But that’s not true. Plenty of 16-year-olds don’t get a license. There are additional conditions besides age: passing the test, being physically able to drive, etc.

Another way of putting that point: being at least 16 years old is not a *sufficient* condition for having a driver’s license; it’s not enough on its own. An example of a sufficient condition: getting 100% on every test is a sufficient condition for getting an A in a class (supposing tests are the only evaluations). That is, if you get 100% on every test, then you’ll get an A. If ‘H’ stands for ‘I got 100% on all the tests’ and ‘A’ stands for ‘I got an A in the class’, then we would indicate that H is a sufficient condition for A in SL by writing  $H \supset A$ . Notice that it’s not a necessary condition: you don’t have to be perfect to get an A.  $A \supset H$  would symbolize a falsehood.

To define a concept is to provide necessary and sufficient conditions for falling under it. For example, a bachelor is, by definition, an unmarried male. That is, being an unmarried male is necessary and sufficient for being a bachelor: you don’t qualify as a bachelor if you’re not an unmarried male, and being an unmarried male is enough, on its own, to qualify for bachelorhood. It’s for circumstances like this that we have the triple-bar. Recall, the phrase that triple-bar is meant to capture the meaning of is ‘if and only if’. We’re now in a position to understand that locution. Consider the claim that I am a bachelor if and only if I am an unmarried male. This is really a conjunction of two claims: I am a bachelor *if* I’m an unmarried male, *and* I’m a bachelor *only if* I’m an unmarried male. Let ‘B’ stand for ‘I’m a bachelor’ and ‘U’ stand for ‘I’m an unmarried male.’ Our claim is then B if U, and B only if U. We know how to deal with ‘if’ on its own between two sentences: the one after the ‘if’ is the antecedent of the conditional. And we know how to deal with ‘only if’: the sentence before it is the antecedent, and the sentence after it is the consequent. To symbolize ‘I am a bachelor if and only if I am an unmarried male’ using horseshoes and a dot, we get this:

$$(U \supset B) \bullet (B \supset U)$$

The left-hand conjunct is the ‘if’ part; the right-hand conjunct is the ‘only if’ part. The purpose of the triple-bar is to give us a way of symbolizing such claims more easily, with a single symbol. ‘I am a bachelor if and only if I am an unmarried male’ can be translated into SL as  $B \equiv U$ , which is just shorthand for the longer conjunction of conditionals above. And given that ‘necessary and sufficient’ is also just a conjunction of two conditionals, we use triple-bar for that locution as well. (Also, the phrase ‘just in case’ can be used to express a biconditional claim.)

At this point, you may have an objection: why include triple-bar in SL at all, if it's dispensable in favor of a dot and a couple of horseshoes? Isn't it superfluous? Well, yes and no. We could do without it, but having it makes certain translations easier. As a matter of fact, this is the case for all of our symbols. It's always possible to replace them with combinations of others. Consider the horseshoe. It's false when the antecedent is true and the consequent false, true otherwise. So really, it's just a claim that it's not the case that the antecedent is true and the conclusion false—a negated conjunction. We could replace any  $p \supset q$  with  $\sim (p \bullet \sim q)$ . And the equivalences we saw earlier—DeMorgan's Laws—show us how we can replace dots with wedges and *vice versa*. It's a fact (I won't prove it; take my word for it) that we could get by with only two symbols in our language: tilde and any one of wedge, dot, or horseshoe.<sup>11</sup> So yeah, we have more symbols than we need, strictly speaking. But it's convenient to have the number of symbols that we do, since they line up neatly with English locutions, making translation between English and SL much easier than it would be otherwise.

## EXERCISES

Translate the following into SL, using the bolded capital letters to stand for simple sentences.

1. Harry Lime is a **C**riminal, but he's not a **M**onster.
2. If **T**horwald didn't kill his wife, then **J**effries will look foolish.
3. Rosemary doesn't love both **M**ax and **H**erman.
4. Michael will not **K**ill Fredo if his **M**other is still alive.
5. Neither **W**oody nor **B**uzz could defeat Zurg, but **R**ex could.
6. If either **F**redo or **S**onny takes over the family, it will be a **D**isaster.
7. **E**li will get rich only if **D**aniel doesn't drink his milkshake.
8. Writing a hit **P**lay is necessary for Rosemary to fall in **L**ove with Max.
9. Kane didn't **W**in the election, but if the opening of the **O**pera goes well he'll regain his **D**ignity.
10. If **D**ave flies into the **M**onolith, then he'll have a **T**ransformative experience; but if he doesn't fly into the **M**onolith, he will be stuck on a **G**host ship.

---

<sup>11</sup> In fact, it's possible to get by with only one symbol: if we defined a new two-place operator that's true when both components are false, and false otherwise, that would do the trick. The symbol typically used for this truth-function is ' $\downarrow$ ', called the "Sheffer stroke" after the logician (Henry Sheffer) who first published this result.

11. Kane wants **L**ove if and only if he gets it on his own **T**erms.
12. Either Henry keeps his **M**outh shut and goes to **J**ail for a long time or he **R**ats on his friends and lives the rest of his life like a **S**chnook.
13. Only if Herman builds an **A**quarium will Rosemary **L**ove him.
14. **K**illing Morrie is sufficient for keeping him **Q**uiet.
15. Jeffries will be **V**indicated, provided Thorwald **K**illed his wife and Doyle **A**dmits he was right all along.
16. **C**ollaborating with Cecil B. DeMille is necessary to **R**evive Norma's career, and if she does not **C**ollaborate with DeMille, Norma may go **I**nsane.
17. Either **D**aniel or **E**li will get the oil, but not both.
18. To have a **F**ulfilling life as a toy, it is necessary, but not sufficient, to be **P**layed with by children.
19. The Dude will get **R**ich if Walter's **P**lan works, and if the Dude gets **R**ich, he'll buy a new **B**owling ball and a new **C**arpet.
20. Either the AE-35 **U**nit is really malfunctioning or HAL has gone **C**razy; and if HAL has gone **C**razy, then the **M**ission will be a failure and neither **D**ave nor **F**rank will ever get home.

## V. Testing for Validity in SL

Having dealt with the task of taming natural language, we are finally in a position to complete the second and third steps of building a logic: defining logical form and developing a test for validity. The test will involve applying skills that we've already learned: setting up truth tables and computing the truth-values of compounds. First, we must define logical form in SL.

### Logical Form in SL

This will seem trivial, but it is necessary. We're learning how to evaluate arguments expressed in SL. Like any evaluation of deductive arguments, the outcome hinges on the argument's form. So what is the form of an SL argument? Let's consider an example; here is an argument in SL:

$$\begin{array}{l} A \supset B \\ \underline{\sim B} \\ \therefore \sim A \end{array}$$

‘A’ and ‘B’ stand for simple sentences in English; we don’t care which ones. We’re working within SL: given an argument in this language, how do we determine its form? Quite simply, by systematically replacing capital letters with variables (lower-case letters like ‘p’, ‘q’, and ‘r’). The form of that particular SL argument is this:

$$\begin{array}{l} p \supset q \\ \sim q \\ \hline \therefore \sim p \end{array}$$

The replacement of capital letters with lower-case variables was *systematic* in this sense: each occurrence of the same capital letter (e.g., ‘A’) was replaced with the same variable (e.g., ‘p’).

To generate the logical form of an SL argument, what we do is systematically replace SL *sentences* with what we’ll call *sentence-forms*. An SL sentence is just a well-formed combination of SL symbols—capital letters, operators, and parentheses. A sentence-form is a combination of symbols that *would be* well-formed in SL, except that it has lower-case variables instead of capital letters.

Again, this may seem like a trivial change, but it is necessary. Remember, when we’re testing an argument for validity, we’re checking to see whether its form is such that it’s *possible* for its premises to turn out true and its conclusion false. This means checking various ways of filling in the form with particular sentences. Variables—as the name suggests—can *vary* in the way we need: they are generic and can be replaced with any old particular sentence. Actual SL constructions feature capital letters, which are actual sentences having specific truth-values. It is conceptually incoherent to speak of checking different possibilities for actual sentences. So we must switch to sentence-forms.

## The Truth Table Test for Validity

To test an SL argument for validity, we identify its logical form, then create a truth table with columns for each of the variables and sentence-forms in the argument’s form. Filling in columns of Ts and Fs under each of the operators in those sentence-forms will allow us to check for what we’re looking for: an instance of the argument’s form for which the premises turn out true and the conclusion turns out false. Finding such an instance demonstrates the argument’s invalidity, while failing to find one demonstrates its validity.

To see how this works, it will be useful to work through an example. Consider the following argument in English:

If Democrats take back Congress, lots of new laws will be passed.  
Democrats won’t take back Congress.  
 ∴ Lots of new laws won’t be passed.

We’ll evaluate it by first translating it into SL. Let ‘D’ stand for ‘Democrats take back Congress’ and ‘L’ stand for ‘Lots of new laws will be passed’. This is the argument in SL:

$$\begin{array}{l} D \supset L \\ \sim D \\ \hline \therefore \sim L \end{array}$$

First, the logical form. Replacing ‘D’ with ‘p’ and ‘L’ with ‘q’, we get:

$$\begin{array}{l} p \supset q \\ \sim p \\ \hline \therefore \sim q \end{array}$$

Now we set up a truth table, with columns for each of the variables and columns for each of the sentence-forms. To determine how many rows our table needs, we note the number of different variables that occur in the argument-form (call that number ‘n’); the table will need  $2^n$  rows. In this case, we have two variables—‘p’ and ‘q’—and so we need  $2^2 = 4$  rows. (If there were three variables, we would need  $2^3 = 8$  rows; if there were four,  $2^4 = 16$ ; and so on.) Here is the table we need to fill in for this example:

p	q	$p \supset q$	$\sim p$	$\sim q$

First, we fill in the “base columns”. These are the columns for the variables. We do this systematically. Start with the right-most column (under ‘q’ in this case), and fill in Ts and Fs alternately: T, F, T, F, T, F, ... as many times as you need—until you’ve got a truth-value in every row. That gives us this:

p	q	$p \supset q$	$\sim p$	$\sim q$
	T			
	F			
	T			
	F			

Next, we move to the base column to the left of the one we just filled in (under ‘p’ now), and fill in Ts and Fs by alternating in twos: T, T, F, F, T, T, F, F, ... as many times as we need. The result is this:

p	q	$p \supset q$	$\sim p$	$\sim q$
T	T			
T	F			
F	T			
F	F			

If there were a third base column, we would fill in the Ts and Fs by alternating in fours: T, T, T, T, F, F, F, F.... For a fourth base column, we would alternate every other eight. And so on.

Next, we need to fill in columns of Ts and Fs under each of the operators in the statement-forms' columns. To do this, we apply our knowledge of how to compute the truth-values of compounds in terms of the values of their components, consulting the operators' truth table definitions. We know how to compute the values of  $p \supset q$ : it's false when p is true and q false; true otherwise. We know how to compute the values of  $\sim p$  and  $\sim q$ : those are just the opposites of the values of p and q in each of the rows. Making these computations, we fill the table in thus:

p	q	$p \supset q$	$\sim p$	$\sim q$
T	T	T	F	F
T	F	F	F	T
F	T	T	T	F
F	F	T	T	T

Once the table is filled in, we check to see if we have a valid or invalid form. The mark of an invalid form is that it's possible for the premises to be true and the conclusion false. Here, the rows of the table are the possibilities—the four possible outcomes of plugging in particular SL sentences for the variables: both true; the first is true, but the second false; the first false but the second true; both false. The reason we systematically fill in the base columns as described above is that the method ensures that our rows will collectively exhaust all these possible combinations.

So, to see if it's possible for the premises to come out true and the conclusion to come out false, we check each of the rows, looking for one in which this happens—one in which there's a T under ' $p \supset q$ ', a T under ' $\sim p$ ', and an F under ' $\sim q$ '. And we have one: in row 3, the premises come out true and the conclusion comes out false. This is enough to show that the argument is invalid:

p	q	$p \supset q$	$\sim p$	$\sim q$
T	T	T	F	F
T	F	F	F	T
F	T	<b>T</b>	<b>T</b>	<b>F</b>
F	F	T	T	T

INVALID

When we're checking for validity, we're looking for one thing, and one thing only: a row (or rows) in which the premises come out true and the conclusion comes out false. If we find one, we have shown that the argument is invalid. If we don't find one, that indicates that it's *impossible* for the premises to be true and the conclusion false—and so the argument is valid. Either way, the only thing we look for is a row with true premises and a false conclusion. Every other kind of row is irrelevant. It's common for beginners to mistakenly think they are. The fourth row in the table above, for example, looks significant. Everything comes out true in that row. Doesn't that mean something—something good, like that the argument's valid? No. Remember, each row represents a *possibility*; what row 4 shows is that it's *possible* for the premises to be true and the conclusion true. But that's not enough for validity. For an argument to be valid, the premises must *guarantee* the conclusion; whenever they're true, the conclusion *must* be true. That it's merely *possible* that they all come out true is not enough.

Let's look at a more involved example, to see how the computation of the truth-values of the statement-forms must sometimes proceed in stages. The skill required here is nothing new—it's just identifying main operators and computing the values of the simplest components first—but it takes careful attention to keep everything straight. Consider this SL argument (never mind what its English counterpart is):

$$\begin{array}{l} (\sim A \bullet B) \vee \sim X \\ \underline{B \supset A} \\ \therefore \sim X \end{array}$$

To get its form, we replace 'A' with 'p', 'B' with 'q', and 'X' with 'r':

$$\begin{array}{l} (\sim p \bullet q) \vee \sim r \\ \underline{q \supset p} \\ \therefore \sim r \end{array}$$

So our truth-table will look like this (eight rows because we have three variables;  $2^3 = 8$ ):

p	q	r	$(\sim p \bullet q) \vee \sim r$	$q \supset p$	$\sim r$

Filling in the base columns as prescribed above—alternating every other one for the column under 'r', every two under 'q', and every four under 'p'—we get:

p	q	r	$(\sim p \bullet q) \vee \sim r$	$q \supset p$	$\sim r$
T	T	T			
T	T	F			
T	F	T			
T	F	F			
F	T	T			
F	T	F			
F	F	T			
F	F	F			

Now we turn our attention to the three sentence-forms. We'll start with the first premise, the compound ' $(\sim p \bullet q) \vee \sim r$ '. We need to compute the truth-value of this formula. We know how to do this, provided we have the truth-values of the simplest parts; we've solved problems like that already. The only difference in the case of truth tables is that there are multiple different assignments of truth-values to the simplest parts. In this case, there are eight different ways of assigning truth-values to 'p', 'q', and 'r'; those are represented by the eight different rows of the table. So we're solving a problem we know how to solve; we're just doing it eight times.

We start by identifying the main operator of the compound formula. In this case, it's the wedge: we have a disjunction; the left-hand disjunct is ' $(\sim p \bullet q)$ ', and the right-hand disjunct is ' $\sim r$ '. To figure out what happens under the wedge in our table, we must first figure out the values of these components. Both disjuncts are themselves compound: ' $(\sim p \bullet q)$ ' is a conjunction, and ' $\sim r$ ' is a negation. Let's tackle the conjunction first. To figure out what happens under the dot, we need to know the values of ' $\sim p$ ' and 'q'. We know the values of 'q'; that's one of the base columns. We must compute the value of ' $\sim p$ '. That's easy: in each row, the value of ' $\sim p$ ' will just be the opposite of the value of 'p'. We note the values under the tilde, the operator that generates them:

p	q	r	$(\sim p \bullet q) \vee \sim r$	$q \supset p$	$\sim r$
T	T	T	F		
T	T	F	F		
T	F	T	F		
T	F	F	F		
F	T	T	T		
F	T	F	T		
F	F	T	T		
F	F	F	T		

To compute the value of the conjunction, we consider the result, in each row, of the truth-function for dot, where the inputs are the value under the tilde in ' $\sim p$ ' and the value under 'q' in the base column. In rows 1 and 2, it's  $F \bullet T$ ; in rows 3 and 4,  $F \bullet F$ ; and so on. The results:

p	q	r	$(\sim p \bullet q) \vee \sim r$	$q \supset p$	$\sim r$
T	T	T	F F		
T	T	F	F F		
T	F	T	F F		
T	F	F	F F		
F	T	T	T T		
F	T	F	T T		
F	F	T	T F		
F	F	F	T F		

The column we just produced, under the dot, gives us the range of truth-values for the left-hand disjunct in the first premise. We need the values of the right-hand disjunct. That's just ' $\sim r$ ', which is easy to compute: it's just the opposite value of ' $r$ ' in every row:

p	q	r	$(\sim p \bullet q) \vee \sim r$	$q \supset p$	$\sim r$
T	T	T	F F F		
T	T	F	F F T		
T	F	T	F F F		
T	F	F	F F T		
F	T	T	T T F		
F	T	F	T T T		
F	F	T	T F F		
F	F	F	T F T		

Now we can finally determine the truth-values for the whole disjunction. We compute the value of the wedge's truth-function, where the inputs are the columns under the dot, on the one hand, and the tilde from ' $\sim r$ ' on the other.  $F \vee F$ ,  $F \vee T$ ,  $F \vee F$ , and so on:

p	q	r	$(\sim p \bullet q) \vee \sim r$	$q \supset p$	$\sim r$
T	T	T	F F <b>F</b> F		
T	T	F	F F <b>T</b> T		
T	F	T	F F <b>F</b> F		
T	F	F	F F <b>T</b> T		
F	T	T	T T <b>T</b> F		
F	T	F	T T <b>T</b> T		
F	F	T	T F <b>F</b> F		
F	F	F	T F <b>T</b> T		

Since that column represents the range of possible values for the entire sentence-form, we highlight it. When we test for validity, we're looking for rows where the premises as a whole come out true; we'll be looking for the value under their main operators. To make that easier, just so we don't lose track of things visually because of all those columns, we highlight the one under the main operator.

Next, the second premise, which is thankfully much less complex. It is, however, slightly tricky. We need to compute the value of a conditional here. But notice that things are a bit different than usual: the antecedent, 'q', has its base column to the *right* of the column for the consequent, 'p'. That's a bit awkward. We're used to computing conditionals from left-to-right; we'll have to mentally adjust to the fact that ' $q \supset p$ ' goes from right-to-left. (Alternatively, if it helps, you can simply reproduce the base columns underneath the variables in the ' $q \supset p$ ' column.) So in the first two rows, we compute  $T \supset T$ ; but in rows 3 and 4, it's  $F \supset T$ ; in rows 5 and 6, it's  $T \supset F$  (the only circumstance in which conditionals turn out false); and in rows 7 and 8, it's  $F \supset F$ . Here is the result:

p	q	r	$(\sim p \bullet q) \vee \sim r$	$q \supset p$	$\sim r$
T	T	T	F F <b>F</b> F	T	
T	T	F	F F <b>T</b> T	T	
T	F	T	F F <b>F</b> F	T	
T	F	F	F F <b>T</b> T	T	
F	T	T	T T <b>T</b> F	F	
F	T	F	T T <b>T</b> T	F	
F	F	T	T F <b>F</b> F	T	
F	F	F	T F <b>T</b> T	T	

No need to highlight that column, as it's the only one we produced for that premise, so there can be no confusion.

We finish the table by computing the values for the conclusion, which is easy:

p	q	r	$(\sim p \bullet q) \vee \sim r$	$q \supset p$	$\sim r$
T	T	T	F F <b>F</b> F	T	F
T	T	F	F F <b>T</b> T	T	T
T	F	T	F F <b>F</b> F	T	F
T	F	F	F F <b>T</b> T	T	T
F	T	T	T T <b>T</b> F	F	F
F	T	F	T T <b>T</b> T	F	T
F	F	T	T F <b>F</b> F	T	F
F	F	F	T F <b>T</b> T	T	T

Is the argument valid? We look for a row with true premises and a false conclusion. There are none. The only two rows in which both premises come out true are the second and the eighth, and in both we also have a true conclusion. It is impossible for the premises to be true and the conclusion false, so the argument is valid.

So that is how we test arguments for validity in SL. It's a straightforward procedure; the main source of error is simple carelessness. Go step by step, keep careful track of what you're doing, and it should be easy. It's worth noting that the truth table test is what logicians call a "decision procedure": it's a rule-governed process (an algorithm) that is guaranteed to answer your question (in this case: valid or invalid?) in a finite number of steps. It is possible to program a computer to run the truth table test on arbitrarily long SL arguments. This is comforting, since once one gets more than four variables or so, the process becomes unwieldy.

## EXERCISES

Test the following arguments for validity. For those that are invalid, specify the row(s) that demonstrate the invalidity.

1.  $\sim A, \therefore \sim A \vee B$
2.  $A \supset B, \sim B, \therefore \sim A$
3.  $A \vee B, \sim A, \therefore \sim B$
4.  $\sim (A \equiv B), A \vee B, \therefore A \bullet B$
5.  $\sim (A \supset B), \sim B \vee \sim A, \therefore \sim (\sim A \equiv B)$
6.  $\sim B \vee A, \sim A, A \equiv B, \therefore \sim B$
7.  $A \supset (B \bullet C), \sim B \vee \sim C, \therefore \sim A$
8.  $\sim A \vee C, \sim B \supset \sim C, \therefore A \equiv C$
9.  $\sim A \vee (\sim B \bullet C), \sim (C \vee B) \supset A, \therefore \sim C \supset \sim B$
10.  $A \vee B, B \supset C, \sim C, \therefore \sim A$