2020

Form and Content: An Introduction to Formal Logic

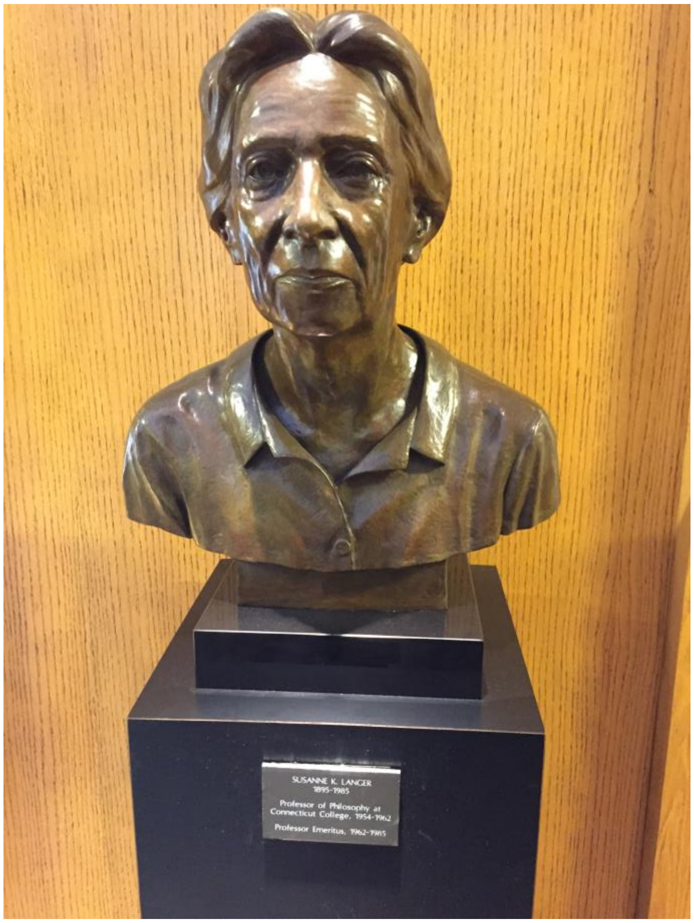
Derek D. Turner
Connecticut College, dtur@conncoll.edu

Recommended Citation
Turner, Derek D., "Form and Content: An Introduction to Formal Logic" (2020). *Open Educational Resources*. 1.
https://digitalcommons.conncoll.edu/oer/1

Form and Content

An Introduction to Formal Logic

Derek Turner
Connecticut College
2020



Susanne K. Langer. This bust is in Shain Library at Connecticut College, New London, CT. Photo by the author.

A Note to Students

For many years, I had relied on one of the standard popular logic textbooks for my introductory formal logic course at Connecticut College. It is a perfectly good book, used in countless logic classes across the country. But the book is expensive. The for-profit publisher routinely makes minor changes—just enough to justify putting out a new edition, which of course costs more. This also creates a dilemma for instructors: In times past, I've had to decide between ordering the soon-to-be out of date edition, which is cheaper and available used, but which students would not be able to sell back, vs. the new edition, which is not available used but which could be resold. A couple of years ago, I lost interest in participating any longer in this sort of system. So I decided to write my own text.

Logic is something that should be accessible to anyone and everyone who takes an interest in studying it. There is something a little unseemly about for-profit textbook publishers making money off of students whose interest in taking a logic course makes them a captive audience. One of my goals in writing this book is thus accessibility. My hope is that this book may serve as an engaging, easy-to-read open access alternative.

This textbook, I'm sure, has lots of flaws and idiosyncrasies. It's written in the first instance for my students at Connecticut College, for the particular sort of logic course that I have developed there over the years. If you see problems, or if there are places where the explanation isn't very clear, please do let me know. Often when you're studying a textbook and you have trouble understanding something, you might be inclined to blame yourself. But please do not do that here. A text can always be improved (especially if no one is trying to make money from it). So as you study this text, I'd like you to read it with a critical eye even as you're trying to learn logic. If you see things that could be done better, let me know, and together we might be able to improve the text for the benefit of future students.

You'll find that most of the examples I use are low-stakes and cheesy—some Harry Potter characters, lots of dinosaurs. My dog, Toby, shows up frequently. I want to say up front that I have reasons for preferring low-stakes examples. Many people, when they first approach logic, are really eager to learn how to apply logic in real argumentative contexts. When reasoning about important issues of the day, how can we spot and diagnose bad arguments? And how can we make better ones? One thing I've found, however, is that when you're trying to learn formal logic, especially for the first time, the content is often a distraction. Many people have strong views about issues like immigration, or abortion, or free speech, or whatever it might be. A logic text that's full of examples involving those and other current matters of political controversy would provoke lots of philosophically rich discussion, but in the course of that discussion the focus would likely drift away from the logic toward the substantive issue that everyone really cares about. However, one of my goals in a logic course is to get you to really care about abstract form—to see it as something glorious and even liberating, as something intrinsically worth studying. I'm convinced that the best way to shine the light on form is to use silly and hopefully fun examples—i.e., silly content—that nobody has much vested interest in. In fact, this is something of a philosophical disagreement that I have with other textbook writers. Everyone wants to know how something like logic might be applicable "in real life," and so other textbooks hit you early with lots of examples involving matters of current interest. Practical applicability really is important, but I'm not convinced that means that it's helpful to have "real life" examples when you're first learning logic. By analogy, if you want to learn how to play clawhammer banjo, it might be best not to start out by trying to play a real song. It might be better to start by spending many hours practicing the right-hand frailing technique, which is really hard to learn but (once you master it) unbelievably graceful and efficient.

The approach that I take in this text—an approach that emphasizes the relationship between form and content—is loosely inspired by the work of Susanne K. Langer, an amazing philosopher and logic aficionado who taught at Connecticut College. Langer also wrote one of the first introductory textbooks on symbolic logic.[1] But she was most famous as a philosopher of art, and especially music. She played the cello, and wrote about music with some authority. Her interest in music was not unrelated to her interest in logic. She saw both as involving the interplay of form and content. Just as you might look at two arguments and find that they both share the same form—*modus ponens*, say—you might play an A minor chord on a piano and a banjo. Same form, different content. Or if you transpose a song from the key of G to the key of A, the form—the relationships among the notes—stays the same, while the content—the actual notes—is changing. Langer understood that there are many different contexts in life in which we might wish to study the interplay of form and content. Logic is just one example of this, where we're interested in the form and content of our own thinking.

One other thing that animates this book is the conviction that technical vocabulary can be empowering. Most people are better at mastering new technical vocabulary than they realize. I do not watch much baseball, but it is always interesting to see how many technical terms the baseball aficionados in my family have mastered: there are terms for positions (short stop, center field, etc.); terms for types of pitches (fastball, slider); terms for different plays (grounding out, double play, stealing second base); statistical terms (at bats, runs batted in, saves), and a host of other technical concepts with precise definitions. Not only that, but baseball aficionados must learn a complicated set of rules (e.g., if a pitcher throws four balls, the batter walks, etc.). Lots of people master this and other similarly complex systems of technical concepts without even realizing it. Without the concepts, you could not understand or appreciate baseball at all. Surprisingly, when it comes to our own reasoning, many people are in the situation of a person watching a baseball game, but without the benefit of any baseball concepts, with no way to process or grasp what is going on. The fact that so few people have the technical logical concepts that would enable them to think and talk clearly about reasoning in the way that baseball fans (for example) can analyze games with great precision is not the fault of anyone in particular. But it is definitely a kind of collective cultural failing. The good thing is that this failing is pretty easy to remedy. The technical concepts of logic are just as easy to master as the technical concepts of baseball.

---

[1] Susanne K. Langer, *An Introduction to Symbolic Logic*, 3rd edition. New York: Dover Publications, 1967. The first edition of the book was originally published in 1937.

And the benefit of studying them is that we no longer have to be alienated from our own reasoning.

There are other open educational resource (OER) logic texts out there. You might also wish to consult with *Forallx*, a wonderful logic text created by P.D. Magnus. One difference between my approach and Magnus's is that I treat semantic issues along the way, rather than treating formal semantics as a distinct topic later in the text. A team at the University of Calgary has also produced a Calgary remix of Magnus's text, which is excellent. Then there is the *Open Logic Text*, which covers more advanced material. I've tried to write this one in a way that's a bit more accessible and conversational. And the order of presentation that I use here is in some ways a little unorthodox. For example, I take a lot of time to explore the notion of formal validity before getting around to the difference between deductive and inductive arguments—an issue which some might think is more basic. (My reason for doing this, in case you are interested, is that I think it's easier to define 'validity' first, and *then* define deduction in terms of validity.) I also try hard along the way to address philosophical questions that in my experience often come up when students are encountering this material for the first time. However, one beautiful thing about open access publishing is that different texts, written in different voices, need not be seen in competition. Seeing the same point explained in different styles can contribute a great deal to learning, so I strongly encourage you to consult other sources. Variety is a good thing.

Thanks

This project was supported by an Open Educational Resources (OER) grant from Connecticut College. I'm grateful to Jessica McCullough and Ariela McCaffrey for their help and support. Mimi Crawford and Ege Sakirt (Connecticut College, Class of '19) both provided invaluable feedback on details when they helped out as course assistants for my introductory logic course at Connecticut College in the Fall of 2017. Evan O'Donnell and Camden Wagner, course assistants in the fall of 2019, have also provided helpful editorial and stylistic suggestions, and helped correct some mistakes. Students in my logic courses at Connecticut College have a helped a great deal with respect to fixing typos and other infelicities.

Outline

Getting Rid of Content

Logical Operators

Truth Tables

Validity and Soundness

How to Prove Stuff

Predicates and Quantifiers

Technical Definitions

The Logic Sheet

Conditional Statements

$p \to q$
if p, then q
p only if q
q if p
p implies q
p is sufficient for q
q is necessary for p
p guarantees q
q is a prerequisite for p

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

p	q	$p \& q$
T	T	T
T	F	F
F	T	F
F	F	F

Conjunction

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

Disjunction

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

Biconditional

p	${\sim}p$
T	F
F	T

Negation

One-way inference rules (introducing operators)

ADD ($\vee$I) $\begin{array}{l} \underline{p\qquad} \\ p \vee q \end{array}$

CON (&I) $\begin{array}{l} p \\ \underline{q\qquad} \\ p \& q \end{array}$

BI ($\leftrightarrow$I) $\begin{array}{l} p \to q \\ \underline{q \to p} \\ p \leftrightarrow q \end{array}$

CP ($\to$I) assume p, derive q, then conclude $p \to q$

One-way inference rules (eliminating operators)

MP ($\to$E) $\begin{array}{l} p \to q \\ \underline{p\qquad\quad} \\ q \end{array}$

DS ($\vee$E) $\begin{array}{l} p \vee q \\ \underline{{\sim}p\quad} \\ q \end{array}$ $\begin{array}{l} p \vee q \\ \underline{{\sim}p\qquad\quad} \\ p \end{array}$

S (&E) $\begin{array}{l} \underline{p \& q} \\ p \end{array}$ $\begin{array}{l} \underline{p \& q} \\ q \end{array}$

BE ($\leftrightarrow$E) $\begin{array}{l} \underline{p \leftrightarrow q} \\ p \to q \end{array}$ $\begin{array}{l} \underline{p \leftrightarrow q} \\ q \to p \end{array}$

Other valid argument forms

MT $\begin{array}{l} p \to q \\ \underline{{\sim}q\quad} \\ {\sim}p \end{array}$

HS $\begin{array}{l} p \to q \\ \underline{q \to r} \\ p \to s \end{array}$

CD $\begin{array}{l} (p \to q) \& (r \to s) \\ \underline{p \vee r\qquad\qquad\qquad} \\ q \vee s \end{array}$

Two-way replacement rules (logical equivalences)

DM	${\sim}(p \& q)$	${\sim}p \vee {\sim}q$
	${\sim}(p \vee q)$	${\sim}p \& {\sim}q$
COM	$p \& q$	$q \& p$
	$p \vee q$	$q \vee p$
Taut	p	$p \vee p$
	p	$p \& p$
Assoc	$(p \& q) \& r$	$p \& (q \& r)$
	$(p \vee q) \vee r$	$p \vee (q \vee r)$
Trans	$p \to q$	${\sim}q \to {\sim}p$
MI	$p \to q$	${\sim}p \vee q$
ME	$p \leftrightarrow q$	$(p \to q) \& (q \to p)$
	$p \leftrightarrow q$	$(p \& q) \vee ({\sim}p \& {\sim}q)$
Exp	$p \to (q \to r)$	$(p \& q) \to r$
Dist	$(p \& q) \vee (p \& r)$	$p \& (q \vee r)$
	$(p \vee q) \& (p \vee r)$	$p \vee (q \& r)$
DN	p	${\sim}{\sim}p$

Can apply to any expression, even part of a line.

An argument is **valid** if and only if it's the case that if the premises are true, then the conclusion is also true.

Rules for the universal quantifier

Universal instantiation (UI)

$$\frac{\forall xFx}{Fa} \qquad \frac{\forall xFx}{Fy}$$

Universal generalization (UG)

$$\frac{Fy}{\forall xFx}$$

Restriction: UG is not allowed inside a conditional proof.

Identity

(=I) $a = a$

(=E)

$$a = b$$
$$\frac{Fa}{Fb}$$

Rules for the existential quantifier

Existential instantiation (EI)

$$\frac{\exists xFx}{Fa}$$

Restriction: EI only works with a new constant that does not already appear in the proof.

Existential generalization (EG)

$$\frac{Fa}{\exists xFx}$$

Equivalence rules for quantifiers

QN	$\sim\forall xFx$	$\exists x\sim Fx$
	$\forall x\sim Fx$	$\sim\exists xFx$
QE	$\forall xFx$	$\sim\exists x\sim Fx$
	$\exists xFx$	$\sim\forall x\sim Fx$

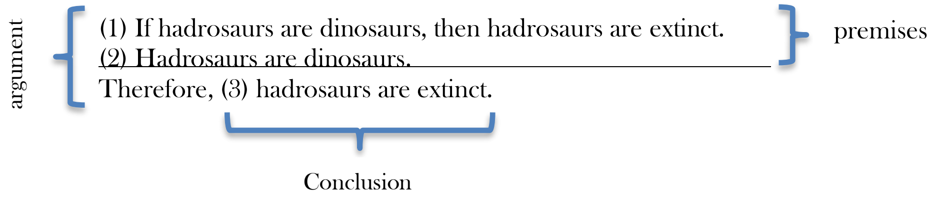
§1. *Modus Ponens*

|Outline|

Susanne K. Langer held that “the science of logic is a steady progression from the concrete to the abstract, from contents with certain forms to those forms without their contents …” (1953, p. 240). There is no better way to begin studying logic than with an example of how this works:

(1) If hadrosaurs are dinosaurs, then hadrosaurs are extinct.
(2) Hadrosaurs are dinosaurs.
Therefore, (3) hadrosaurs are extinct.

What is this, exactly? At its most basic level, this is just a set of statements. But the statements are also ordered in a certain way. The word “therefore” suggests that statements (1) and (2) are meant to justify, or provide some reason for believing statement (3). So let’s call this an argument. An argument, let’s say, is a set of statements, some of which are meant to provide support or justification for one of the others. The statement that’s supposed to get the justification is the conclusion. In this case, the conclusion is statement (3). The statements that are meant to provide reasons for believing the conclusion are the premises. In this case, the premises are statements (1) and (2).



In what follows, we will be focusing a lot on arguments with two premises. However, arguments can have only one premise. And they may also have many more than two premises.

We can do a little more analyzis of this example. Notice that statement (1) has an “if … then …” structure. This is an example of a conditional statement, or a hypothetical statement. Every conditional statement has two other statements as components. These are highlighted here:

(1) If hadrosaurs are dinosaurs, then hadrosaurs are extinct.

The reason for calling this a conditional statement is that it says that the green part is conditional upon the yellow part. Notice that each component is a statement in its own right. The one on the left, highlighted in yellow, is the antecedent of the conditional statement. The one on the right, in green, is the consequent. A conditional statement is a good example of a complex statement (also sometimes called a compound statement). In general, a complex statement is any statement that contains at least one other statement as a part. A simple statement is one that contains no other statements as parts. For example, “Hadrosaurs are dinosaurs” is a simple statement.

In addition to the antecedent and the consequent, a conditional statement has what is known as a logical operator (sometimes also called a logical connective). In this case, the logical operator is simply the “If … then …” In general, a logical operator is a device that lets us take simple statements and form complex statements out of them.

If you read some older logic texts or papers, you might find logicians talking about “atomic” statements *vs.* “molecular’ statements. This is just a bit of old-fashioned terminology, inspired by chemistry. An atomic statement is just a simple one. The original meaning of “atom,” derived from ancient Greek, was something that cannot be divided into parts. So a simple/atomic statement is one that cannot be divided into other statements. A molecular statement, like a molecule, is one that’s built up from simple statements.

The argument above is also a lovely example of a type or form of argument known as *modus ponens*. (We’ll see that many types of arguments have Latin names. In this case, modus ponens is actually short for *modus ponendo ponens*, which is Latin for “mode that affirms by affirming.”) In a *modus ponens* argument, one premise is always a conditional statement. The other premise just asserts or affirms the antecedent of the conditional. For this reason, you could also think of modus ponens as *affirming the antecedent*. The conclusion asserts the consequent. The highlighting below may help to make the structure a little clearer:

(1) **If** hadrosaurs are dinosaurs, **then** hadrosaurs are extinct.
(2) Hadrosaurs are dinosaurs.
Therefore, (3) hadrosaurs are extinct.

Logicians treat arguments as objects of study, just as a botanist might study plants. In order to get a better understanding of plants, you need some technical vocabulary, so that you can classify and describe them with greater clarity. The same is true in logic. One of the challenges of studying logic is acquiring the technical terminology that will enable you to analyze arguments. That technical terminology is also empowering, and will help you to see things that you never appreciated before. We’ll spend quite a bit of time mastering some of the basic technical concepts of logic. So it will take a while before we really enjoy the payoff.

Technical Terms

- argument
- premise
- conclusion
- conditional statement
- antecedent
- consequent
- simple *vs.* complex statements
- logical connective
- logical operator

§2. Abstraction and Form

|Outline|

The distinction that's at the very heart of logic is that between the *form* and the *content* of reasoning. One of the goals of studying logic is to become adept at perceiving form. This takes a lot of practice, but once you get good at it, you may find that you have developed a kind of perceptual superpower. Interestingly, learning the vocabulary of logic will actually help you become a better perceiver of logical form. Being able to give a name to a certain formal structure is crucial to being able to recognize it when you see it. For example, when a geologist learns the technical vocabulary for different types of rocks—schist vs. gneiss, and all that—it can help them see the difference between those rocks. Much the same is true with art forms such as music or film. Learning the technical vocabulary of film theory—*mise en scéne*, diagetic sound, etc.—can help you to see things on screen than you never appreciated.

Consider the following two examples:

(1) If hadrosaurs are dinosaurs, then hadrosaurs are extinct.
(2) Hadrosaurs are dinosaurs.
(3) Therefore, hadrosaurs are extinct.

(1) If Toby goes to the park, then Skipper stays home.
(2) Toby goes to the park.
(3) Therefore, Skipper stays home.

(Toby is my dog, and Skipper is one of his friends.) These two arguments are about completely different things—dinosaurs *vs.* dogs at the park. Or to put it another way, they have entirely different subject matter or *content*. But the formal structure of both arguments is exactly the same. Both are instances of *modus ponens* (§1). But this fact can be hard to see. Often, when we're thinking or conversing with others, we're more focused on the subject matter than anything else. After all, the subject matter is what we usually care most about. However, as you begin to study logic, you may begin to find the *form* of reasoning more and more interesting. You may even get to the point—I hope you do!—where the form of reasoning strikes you as something inherently beautiful and fascinating.

It's helpful to have techniques for bringing out the form of an argument and making it visible. One way to do this is simply to use letters:

(1) If *p*, then *q*
(2) *p*
(3) Therefore, *q*

Here the lower-case italicized letters '*p*' and '*q*' are just placeholders or statement variables. Those letters do not stand for any statement in particular. The letter '*p*' just means, "some statement or other." Using variables in this way helps to bring out what the two arguments have in common—namely, their form. "*Modus ponens*," again, is just the name commonly given to that form. Depending on what statements you plug in for the variables, you could construct a *modus ponens* argument about any topic under the sun. You can hold the form fixed while varying the content endlessly. I should mention that in many other textbooks, you will see logicians using Greek letters for these statement variables. Where I wrote, "If *p*, then *q*," others might say: "If *φ*, then *ψ*." The Greek letters make things seem a little more mathematical and intimidating, but they serve little other purpose.

Logic is not the only place where we might hold the form fixed while varying the content. Another great example of this is playing the same song in different keys. Imagine, for example, that you have a simple two chord song in the key of G. So you have your G chord and your D chord, let's say. (The G is the base, or I chord, and the D is the V chord.) It's easy to transpose the song into a new key, say the key of A. Then your base chord would be A, and the original D chord would get replaced by the E. As you switch from the key of G to the key of A, the structure or form of the song remains the same, the actual notes are all different.

One skill that we cultivate when studying logic is abstraction. Abstraction is important in many different fields, but it has a special pride of place in logic. It is, in a way, what logic is all about. It is the setting aside of content or subject matter in order to study and appreciate form, pattern, or structure. It is, in effect, what we did just now when we started with two concrete arguments—an argument about dinosaurs and one about dogs at the park—and noticed that they have the same form. In order to recognize that form and give it a name—*modus ponens*—we have to abstract away from the topics that the arguments are, in some sense, about. Interpretation is the opposite of abstraction. To interpret a formal structure is to give it content.

It helps to have a way of talking about the relationship between form and content when it comes to reasoning. So we need some additional terminology. Let's call the following an argument form:

(1) If *p*, then *q*
(2) *p*
Therefore, (3) *q*

And then we can say that the two arguments with which we started are substitution instances of this form. They are, in other words, substitution instances of *modus ponens*. What's the difference between a mere argument form and an actual argument? To start with, an actual argument has to have *both* form and content. An argument form is just the form without the content. But there's another way to see the difference. In §1 we defined an argument as a special kind of set of statements. But if you think carefully about the argument form above, you'll see that it's not really a set of statements at all. Consider line (2), which just says '*p*'. '*p*' is not a genuine statement. It's just a variable or an empty placeholder.

On a historical note, the ancient Greek stoics were the first thinkers to identify this sort of argument form, including *modus ponens* and some other argument forms that we will look at later. The stoic philosopher Chrysippus was one of the great logicians of the ancient world. The only person who comes close to Chrysippus was Aristotle, who also made huge contributions to logic, while adopting a completely different approach. The stoics are probably best known for their seemingly crazy views about the good life. They argued that moral virtue, rationality, and happiness are all the same thing. They prioritized the study of logic in part because they saw greater rationality as the path to happiness.

Technical Terms

- statement variables
- abstraction
- interpretation
- argument form
- substitution instance

§3. Use and Mention

|Outline|

Consider the difference between the following statements:

(1) Toby has four letters.
(2) 'Toby' has four letters.

Statement (2) is obviously a statement about the name 'Toby.' In statement (2), the name 'Toby' is being **mentioned**, rather than used to refer to anyone. The single quotation marks signal the mentioning here. (In general, I will use single quotation marks when mentioning a term or statement; I'll save double quotation marks for actual quotations). Statement (2) is boring, and obviously true. But statement (1) is a little funny. There, the name 'Toby' is being **used** to refer to someone—my dog. In order to make any sense of statement (1), we have to imagine a context where it might make sense for a dog to be collecting letters. Maybe Toby is playing with Scrabble pieces on the living room floor. That is one context in which statement (1) might be true.

In ordinary life, many people are very sloppy when it comes to the difference between use and mention. One reason for the sloppiness, perhaps, is that in spoken conversation it's very difficult to mark the difference between use and mention. If you read statements (1) and (2) out loud, they would sound the same, although you could perhaps use air quotes when reading statement (2).

The distinction between use and mention is closely related to another distinction between the object language and the meta-language. Imagine that you are taking a German class, and the instructor says something like the following:

The past tense of 'Wir haben' is 'Wir hatten.'

In this case, the teacher is using English to say something about German. The German expressions are being mentioned, but not used. Of course, you *could* use the German expressions to say something, but then you would be speaking German! The language you are saying something about—in this case, German—is the **object language**. The object language is, in a sense, the object or topic of conversation. The language that you are using to express yourself—in this case, English—is the **metalanguage**.

Sometimes, one and the same language can be both the object and the meta-language. For example, it's easy to use English to talk about English.

The past tense of 'We have' is 'We had.'

Here, English is both the object and the meta-language.

There are different ways of thinking about what logic is all about. I suggested above (§2) that it's largely about the *form* of reasoning. Another way to think about it is as the study of certain **artificial languages**. An artificial language, as contrasted with a **natural language** (like English or German), is just a symbol system that we have invented for some purpose. Computer programming languages are also examples of artificial languages. As we go along, it will be really helpful to think of ourselves as constructing an artificial language from scratch. That language will be “classical logic,” but we may as well give it a name that’s more fun than that. I propose to call it “Langerese,” in honor of Susanne K. Langer, who taught at Connecticut College, and who wrote one of the very first introductory logic texts. So for much of what follows, Langerese will be our object language, while English is our meta-language, the language we use to talk about Langerese.

One issue with the use/mention distinction is that if you take great care to track when you are merely mentioning a term or expression, you will end up using lots of quotation marks. This can be a bit of a pain. Sometimes, when it’s clear from context whether I am using vs. mentioning something, I may (in what follows) get lazy and ignore the quotation marks. Just to warn you, I am especially likely to do this when talking about individual sentence letters. For example, I might say things like: “Suppose we assign to *P* the truth-value T.” Technically, the sentence letter ‘*P* ’should be in scarequotes there: “Suppose we assign to ‘*P*’ the truth-value T.” In a case like this where there isn’t much risk of misunderstanding, I may be a bit sloppy and leave out the quotation marks.

Technical Terms

- Use vs. mention
- object language
- meta-language
- artificial language
- natural language

§4. Statements, Statement Forms, and Bivalence

|Outline|

In §1, we defined an argument as a set of statements, one or more of which is meant to provide justification or support for one of the others. But what exactly is a statement? Without knowing what a statement is, we still won't have a very good definition of 'argument.' So here goes. A statement is any sentence that has a truth value. 'True' and 'false' are examples of truth values, but there could be other truth values in addition to those two (more about that later). For the moment, it is accurate enough to think of the truth value of a statement as its truth or falsity.

Consider the difference between the following two expressions:

(1) If hadrosaurs are dinosaurs, then hadrosaurs are extinct.
(2) If *p*, then *q.*

The first of these two expressions is a statement, because it can be true or false. It is, if you will, a declarative sentence. Questions and instructions, by contrast, are not statements at all. (2) looks like a statement; indeed, it has the same 'if … then …" structure as (1). But (2) is merely a statement form, because it has those two variables. If you were to read it out loud to yourself, it might go like this: "If some statement or other, then some statement or other." There's not even any requirement that the statement you plug in for '*p*' has to be different from the one you plug in for '*q*'. Statement (1) is a substitution instance of statement form (2), but so is "If Harry Potter is a wizard, then Harry Potter is a wizard."

Just as an argument form is what you get when you start with an argument and abstract away from the content or subject matter, a statement form is what you get when you remove the content of a statement, leaving only the form. So statement (1) has both form and content. But if you set aside the content—or what the statement is about—you're left with something like pure conditionality, or statement form (2).

Let's use lowercase italicized letters as statement variables: '*p*,' '*q*,' '*r*,' …. If something is to be a bona fide statement, then it must not contain any variables at all. For example:

(3) If hadrosaurs are dinosaurs, then *q*.

This is merely a statement form, not a statement. One more interesting detail: As a matter of principle, there is no limit to how long a statement form can be. Here is a statement form with three variables:

(4) If *p*, then (if *q*, then *r*).

This is a conditional statement form whose consequent is another conditional statement form. If we wanted to, we could keep on using this device to create ever longer statement forms. With only 26 letters in the alphabet, we might eventually run out of letters. If that happens, the best approach is to use subscripts: ‘p_1,’ ‘p_2,’ and so on.

We defined a statement as any sentence that has a truth value. According to the principle of bivalence, every statement is either true or false. There are, in other words, only two truth values. The principle of bivalence is not trivial, although it might be hard to see at first why anyone would question it. There might be cases, though, where we’d want to say that a statement is *neither true nor false*. The principle of bivalence doesn’t allow this. Every statement must be one or the other.

One sort of example that might motivate logicians to drop the principle of bivalence involves vagueness. A term is vague when it’s not entirely clear just which things the predicate applies to. One classic example of this is baldness. Everyone knows people who clearly are not bald. And probably everyone knows people who definitely are bald. But there are borderline cases. Picture a man with a receding hairline. If the hairline has receded far enough, it might be hard to say if he is bald. Maybe he’s sort of bald. Sometimes it’s just not totally clear whether a person should count as bald. Suppose the man in question is named Beauregard. Is the statement “Beauregard is bald” true or false? It’s tempting to say neither, or something in between, or “sort of true.” But the principle of bivalence requires a clear-cut answer here: true or false, and nothing in between.

Some logicians are so concerned about vagueness and other philosophical problems that they drop the principle of bivalence. This opens the door to alternative systems of logic, such as three-valued logic, which allows a third truth value in addition to true and false, which is usually given as just ‘i’, which we could take to stand for indeterminate. We could, however, give whatever name we want to the third truth value. For example, we could have a system with three values: True, False, and Meh. In three-valued logic, the statement that Beauregard is bald might be neither T nor F, but i, or Meh, or whatever.

Fuzzy logic is even more complex, swapping out truth values for numerical values between zero and one. In effect, fuzzy logic quantifies degrees of truth; it’s a system of logic that allows for shades of truth. Both fuzzy logic and three-valued logic involve alterations to basic principles of classical logic, such as the principle of bivalence. In this course, we’ll focus on mastering classical logic (or the artificial language of Langerese) first. It’s easier to wrap your mind around non-classical logic if you first have a good grounding in classical logic. I should mention that non-classical logics such as three-valued logics are sometimes called “deviant logics,” though that is kind of an unfortunate term, given the negative connotations of “deviance.”

It’s important to be aware that many features of classical logic are controversial. It is tempting to think that logic just is the way it is, that it has some kind of special unchangeable purity, that it’s sacrosanct. Some people do feel this way about classical logic, but everything in this context is controversial. In a way, what we call classical logic, which is only 150 years or so old, became “classical” as a result of a social decision by philosophers and logicians. To be totally honest, many aspects of classical logic, including the principle of bivalence, are optional. “Core logic” or even “default logic” would probably be better terms than “classical logic,” since “classical” to many people evokes ancient Greece and Rome.

So why exactly would anyone stick with bivalence? The best answer is that it’s simple. One of the great advantages of classical logic is its amazing elegance and simplicity. It’s streamlined, and easy to work with. If we were to drop bivalence, things would get really complicated, really fast. Bivalence has pros and cons. On the pro side is simplicity. On the con side, it may force us to say weird and implausible things about issues like vagueness. This in itself is philosophically interesting. You might wonder: Why should we care about simplicity here? Is simplicity a kind of aesthetic value? It might be, but perhaps that is as it should be, for the study of forms is a profoundly aesthetic undertaking.

Technical Terms

- statement
- statement form
- truth value
- bivalence
- vagueness
- classical logic
- three valued logic
- fuzzy logic

§5. Statements, Propositions, and Truth

|Outline|

Often, the system of logic that we are beginning to get acquainted with here is called "propositional logic." In presenting things here, I've made a decision to focus on statements (which are one type of sentence) rather than propositions. Nothing much rides on this, but it's worth taking a moment to think about the relationship between statements and propositions. Consider the difference between the following two sentences:

(1) I kicked my brother.
(2) My brother was kicked by me.

Both of these are statements—that is, they are declarative sentences that could be true or false. The only difference between them is grammatical: (1) is in the active voice, while (2) is in the passive voice. But there is an important sense in which (1) and (2) *say exactly the same thing*. We might say that they have the same meaning, or—using some traditional philosophical terminology—that they express the same proposition. So what we have here is two different statements, but only one proposition. A **proposition**, we might say, is the meaning of a statement.

Two statements in different languages can also say the same thing:

(3) It's snowing.
(4) Es schneit.

Statement (3) is in English, while (4) is in German. (3) and (4) are obviously different statements, but they mean the same thing, or as we could say, they express the same proposition.

So what difference does it make whether we think of what we're doing as developing a statement logic vs. a propositional logic? For most practical purposes, this difference won't matter much at all. But I do think there might be one interesting (if not decisive) reason for focusing on statements rather than propositions. That reason has to do with translation. If we treat Langerese, our formal logical language, as an artificial language, then we'll want to be translating English statements into statements of Langerese, much as we would translate from one natural language to another, such as from English to German. But that requires thinking of Langerese as involving statements. If on the other hand, we decided to think of logic as having to do with the structure of propositions, it would not make much sense to treat Langerese as an artificial language. Instead, we'd have to think of Langerese as having to do with the structures of propositions—which is to say, of the things expressed by the statements in various languages. In what follows, then, I will treat Langerese as an artificial language, and will continue to talk about statements in Langerese.

Both statements and propositions are examples of what philosophers sometimes call **truth bearers**. A truth bearer is anything under the sun that could be true or false. In addition to statements and propositions, this includes beliefs, assertions, claims, hypotheses, and other similar sorts of things.

We should note before moving on that there is a huge philosophical issue lurking in the background of our exploration of logic: What is the nature of truth? Philosophers have had many different things to say about what makes a statement true or false. One very traditional view is the correspondence theory of truth, which says (roughly) that true statements agree with reality, while false statements don't. But the correspondence theory can also lead us down various philosophical rabbit holes. For example, what exactly does "agreement with reality" amount to? Also, what about statements such as "2 + 2 = 4"? What is the nature of the mathematical reality with which they might agree or disagree? How might the correspondence theory of truth apply to statements about the future? Philosophers have also developed rival theories, such as the coherence theory of truth, which says that what makes a statement true is its belonging to a coherent set of statements. As you can probably imagine, the big challenge for such a theory is to give a good definition of 'coherence'. These are just two of many options. The literature on theories of truth is quite large, and everything about the topic is philosophically controversial. Interestingly, for the purpose of developing our logical language, Langerese, we can safely set aside these questions about the nature of truth. For logical purposes, it matters *that* there are two truth values, true and false, but it doesn't really matter much *what* these things are. We could even flip things around and think of classical logic as imposing a constraint on theories of truth. Whatever theory of truth you think you want to defend, it had better conform to the rules of classical logic. We might think that logic tells us a lot about how truth values behave, but in a way that's neutral with respect to different theories about the nature of truth.

If you survey other logic textbooks, you might find some that use 0 and 1 instead of T and F. The use of 0 and 1 sometimes reflects an interest in computer science, where binary code serves as the foundation for pretty much everything. But 0 and 1 are just numbers that do not have the same meaning as "true" and "false"—even if the meaning of "true" and "false" is complicated by the fact that philosophers disagree about the nature of truth. Does it matter whether we use T and F as opposed to 0 and 1. This is actually another interesting place where the form/content distinction comes into play. From a more abstract, formal perspective, it simply does not matter at all what our truth values mean. The only thing that matters is that there are two of them, and that they behave in certain ways. However, if we are worried about what logic is *about*, then we might think we need to say something about what the truth values actually mean. And then it might start to matter whether we are using T and F rather than 0 and 1.

Technical Terms

- Proposition
- Truth bearer
- Correspondence theory of truth
- Coherence theory of truth

§6. *Modus ponens* arguments in Langerese: Syntax

|Outline|

We started, in §1, with this simple instance of a *modus ponens* argument:

(1) If hadrosaurs are dinosaurs, then hadrosaurs are extinct.
(2) Hadrosaurs are dinosaurs.
Therefore, (3) hadrosaurs are extinct.

We've already seen how this is an instance of a certain argument form (*modus ponens*), and how statement (1) is an instance of a certain statement form (the conditional). We're now going to embark upon the process of inventing our own artificial language—Langerese. Langerese will be a formal language, in the sense that its main purpose is to bring into focus the forms of reasoning. In order to begin to develop Langerese, we need some symbols for statements, and also a symbol for the "if … then …" operator. Here is a first stab at Langerese translation of the above argument:

(1) $D \rightarrow E$
(2) D
(3) E

We saw earlier how to use statement variables ('p', 'q', etc.) as empty placeholders to help portray statement forms and argument forms. But what we have here is not an argument form. It is, rather, a genuine argument. It's just that the argument is written in Langerese rather than English. It is a Langerese translation of the argument about hadrosaurs. The upper-case, italicized letters that appear here are not variables at all, but rather statement constants. As with variables, if we ever need to use more than 26 letters, we can just use subscripts ('A_1', 'A_2', ….). What makes them statement constants is that we assign a statement to each letter. The letter stands for that statement and that statement alone, so that its meaning remains constant. By contrast, a statement variable is one that stands in for any statement whatsoever, and for none in particular. The arrow symbol ('→') represents the conditional, or roughly, the "if …then … " It tells us that (1) is a conditional statement.

Every language, whether natural or artificial, has both a semantics and a syntax. Langerese is no exception, and we need to be careful to develop both of these components as we go. Semantics is the theory of meaning, whereas syntax is the set of grammatical rules that dictate what does and does not count as a formula of the language. A well-formed formula, or wff (pronounced "woof"), is a formula—that is, just a string of symbols—that is grammatically well-constructed, or grammatically correct. Even in a natural language, we can distinguish between grammatically correct formulas and grammatical failures:

(1) Toby played with his friends at the dog park.
(2) Toby at played with park his friends the.

Statement (1) is a wff in English, whereas (2) is a grammatical train wreck—not a wff at all. Similarly:

(3) $D \rightarrow E$
(4) $D\,E \rightarrow$

In Langerese, (3) is a wff, whereas (4) is not. As a matter of principle, we should be able to write out the grammatical rules for any language. Of course, for a natural language such as English, that's a complicated prospect. One advantage of constructing an artificial language is that it's fairly straightforward to state the grammatical rules. We can start with something along the following lines:

[S1] Every statement letter is a wff.
[S2] If 'p' and 'q' are wffs, then '$(p \rightarrow q)$' is a wff.
[S3] '$pq\rightarrow$' and '$\rightarrow pq$' are not wffs, even if 'p' and 'q' are wffs.

[S1] through [S2] are just examples of the kind of syntactical or grammatical rules that we need to tell us what counts as a well-formed formula (or wff) in Langerese. Taken together, [S1] through [S3] explain why (3) is a wff in Langerese but (4) is not. To make just a few observations about these rules: Note, first, that none of them are stated in Langerese itself; all are stated in the meta-language, which is to say, in English. Second, note the use of statement variables again. This gives us flexibility, since it means that we can plug 'D' or 'E' or any statement constant—or for that matter, any wff at all!—in for either of the variables. Finally, these grammatical rules are recursive. Rules [S1] and [S2] tell us that '$D \rightarrow E$' is a wff. But because it's a wff, we can also plug it in for, say 'q'. Suppose we plug in 'E' for 'p'. Then we get a more complex formula:

(5) $E \rightarrow (D \rightarrow E)$

Recursivity just means that once we form a complex statement according to the grammatical rules, we can re-apply those same rules to form an even more complex statement. Note that statement (5) still has the basic form of a conditional statement. It's an instance of the statement form '$p \rightarrow q$'. Hopefully the green and yellow highlights make it clear what is being substituted for what.

There's one additional detail in statement (5) that deserves mention. Statement (5) has two arrows, or two logical operators, but only one of them is the main operator. Every complex statement has exactly one main operator. Generally speaking, the main operator is the one that governs the entire statement. The arrow on the right is not the main operator because it only governs one part of the statement—namely the consequent, or the part being substituted in for '*q*'. The parentheses are just a handy punctuational device that makes it easier to keep track of what is being substituted for what. In general, when substituting in a wff that's already a complex statement, one should use parentheses or brackets to keep track of which operator is the main one.

From here on out, I will not bother stating the syntactical rules of Langerese explicitly. They are pretty boring, and once you get the idea, you do not need to keep referring to them. (By analogy, once you become a competent speaker of a natural language, you don't necessarily have to keep the grammatical rules before your mind at all times. You might not even remember what the rules are.) But one last consequence is worth mentioning: The rules as I have stated them—even just [S1] through [S3]—permit the construction of ridiculously long wffs. That's because they are recursive. Of course, there are practical and psychological limits on how long a wff any one of us could think about or write down. The point is just that there are *no grammatical or syntactical limits* on how long a wff can be. However, no matter how absurdly long our wffs get, they will always still be finite in length. The reason for that has to do with our recursive definition of wffs. The only way to form a wff under a rule like [S2] above is to start with two wffs, which are already finite, and hitch them to each other with a logical connective. But when you do that, you will always still end up with a wff that's finite in length.

One more technical detail needs a little bit of attention. Often, when we write a statement with only one operator, such as:

$$D \rightarrow E$$

We don't bother putting parentheses around it. Technically, though, '$D \rightarrow E$' is not a wff. Rule [S2] above says that the grammatically correct wff would be:

$$(D \rightarrow E)$$

Above, however, we saw that the main purpose of the parentheses is to help keep track of the main operators. But in this case, where there's only one operator, we know that the arrow is the main one. So although the parentheses are grammatically correct, we don't really need them. If we want to, we can add a new grammatical rule to address this:

> [S4] Where a wff has only one logical operator, we can drop the parentheses and still have a wff.

This little rule will make writing out logical translations a bit easier.

One annoying thing about logic is that things are not quite as standardized as you might expect. Here I've decided to use the '→' symbol for conditional statements. But if you survey other textbooks, you might also find some that use the horseshoe symbol '⊃' instead. In the 1920s, the great Polish logician Jan Lukasciewicz devised another way of writing out logical formulas, an approach that is sometimes called "Polish notation." In Polish notation, the logical operators are always written as prefixes. So '$A \rightarrow B$' would get written as '*CAB*' where the letter '*C*' just stands for conditional. These differences are merely notational. They turn out not to make much difference. However, these superficial decisions about what notation to use do affect what counts as a wff.

The reflections in this section raise some deeper philosophical questions about logic, right here at the beginning. We won't be able to answer those questions in a satisfactory way here, but it's important to flag them. Above I said that Langerese is an artificial language. This contrasts with natural languages, such as English or Spanish. There are some huge differences. So for example, Langerese is merely a written language. You can't *speak* Langerese. You can say things like "*p* arrow *q*," but in that case you're still using the metalanguage—English—to talk about an expression in Langerese. In this respect, it's a lot like computer programming languages (also great examples of artificial languages). However, the most glaring differences between artificial languages and natural languages is that nobody sat down and invented English or Spanish. The rules of natural languages are, perhaps, in some sense, and to some degree, up to us. Think about debates concerning the gender-appropriateness of different English pronouns. But artificial languages are still very different. Since we're building Langerese from scratch, we can, in effect, write the rules any way we want! This is hugely important. As I guide you through the process, I will generally construct the language in the usual way that logicians do. Langerese, at the end of the day, is just classical logic, perhaps with a few of our own accents here or there. But at any point along the way, there will be other directions one could take, other decisions about how to write the rules, how to set things up. Our adoption of the principle of bivalence back in §4 was another good example of such a decision. Nor is it acceptable, really, to ask students to go along with a certain way of doing things just because that's how people always do it. These decisions about how to construct the language require some justification. It's totally reasonable, at any point along the way, to ask: why in the heck do we have to do things like this, rather than some other way?

In most cases, I think there are pretty good justifications for the decisions that go into constructing classical logic. However, none of these justifications are totally decisive, either. It might help to think of an artificial language as a tool that you construct to do a certain job. What kind of tool you need depends on your goals, and on the job that you need to complete.

Technical Terms

- Semantics
- Syntax
- Statement constant
- Main operator
- Well-formed formula (wff)
- Recursivity

§7. *Modus Ponens* Arguments in Langerese: Semantics

|Outline|

At this point, we've introduced the arrow operator as well as the idea of using italicized capital letters to stand uniformly for particular statements. We've also got a few grammatical rules that tell us what does and doesn't count as a well-formed formula of our new language, Langerese. So this facilitates a translation of the argument about hadrosaurs:

(1) $D \rightarrow E$
(2) $\underline{D\qquad\qquad}$
(3) E

But every language needs a semantic theory as well as a grammar. We need to ask, right up front, what these symbols of Langerese *mean*. Or to put it another way, we need to say something about how to interpret the symbols.

The statement letters, or individual constants, are the easy part. If we wanted to, we could say that the meaning of each letter is just the statement that it stands for. So for example, we could assign meaning in the following way:

'*D*': 'Hadrosaurs are dinosaurs.'
'*E*': 'Hadrosaurs are extinct.'

We can call this an interpretation of the statement constants. A second approach is to assign a truth value to each letter. For example, since we know that Hadrosaurs really are dinosaurs, we can assign 'true' to '*D*'. Since hadrosaurs are extinct, we know that '*E*' is also true.

'*D*': true
'*E*': true

These two ways of thinking about how to interpret the statement constants are closely related, and it won't matter much which approach we adopt. After all, which truth value we assign to the letter would seem to depend on which statement it stands for. And once we figure out which statement it stands for, that statement's truth value—whatever it might be—will automatically attach to the letter. Things will be slightly more elegant, though, if we adopt the second approach, and say that an interpretation of a statement constant assigns a truth value to it. Having said that, we will often assign a truth value to a statement constant *by way of* assigning an English statement to it.

That was the easy part. Now what about the '→'? It's extremely tempting to say that '→' just means 'if … then …' But it's really important to resist that temptation. If you can understand *why* it's important to resist that temptation, then you are well on your way to seeing what formal logic is all about!

One beautiful thing about propositional logic—or statement logic, if you prefer to call it that—is that it is truth functional. What this means is that the truth value of any complex statement is uniquely determined by the truth values of its components. So far, the only type of complex statement we have at our disposal is the conditional statement. Langerese is (so far) an extraordinarily boring language consisting only of simple statements and complex conditional statements. Truth functionality just means that if you know the truth values of '*p*' and '*q*', then you know exactly what the truth value of '*p* → *q*' must be. It's really up to us to decide whether Langerese is going to be a truth-functional language. Since we're creating it, we can set it up in whatever way we want. And truth-functionality has some costs and limitations, as we'll soon see. But it also brings great power and elegance. So let's provisionally say that Langerese is going to be truth functional

In general, a function is any operation that takes certain things as inputs and generates outputs in a systematic way. A *truth* function is a function that takes certain truth values as inputs and generates other truth values as outputs. If we wanted to, we could set up any truth function that strikes our fancy. For example, let's have some fun and invent a happy function. This function, which we can represent using the smiley face operator, '☺*p*'. Let's stipulate that the happy function makes any statement true, no matter what its starting truth value might be. If '*p*' is true, then '☺*p*' is true. But if '*p*' is false, then '☺*p*' is still true! The happy function is relatively useless, but it is still a perfectly good example of a truth function. We can display or show how the function works using a very simple table—what's known as a truth table:

Input	Output
p	☺*p*
T	T
F	T

Crucially, a truth table displays all possible inputs, and correlates each possible input with exactly one output. The truth value of '☺*p*' is uniquely determined by the truth value of '*p*'

Now suppose someone asks you what the smiley face operator *means*. What should you say? It won't really help much to try to answer this question by giving an English translation, because there just isn't anything in English that corresponds to '☺' at all. We could give a longer English description of what this silly operator does. That is, we could just say what I said above—that this operator generates an output of 'true' given any input. But we can also *show* this using the truth table above. The truth table is so clear and perspicuous that we may as well say that the truth table just is the definition of '☺'. The truth table, in other words, gives us our semantic theory for the logical operator. Indeed, as long as Langerese remains truth functional, truth tables will serve perfectly well as our semantic theory. This is actually one of the main benefits of truth functionality. Truth tables are amazingly clear and easy—except that they are sometimes counterintuitive.

But before we proceed, we have to tackle one fascinating theoretical question. There are lots of possible truth functions, but we might not want to include *all* of them in Langerese. Should our artificial language include the smiley face operator? The smiley face operator is nonstandard, and is not typically used. But why not? How should we go about determining which truth functions deserve operators in our language and which should be ignored?

One problem with the smiley face operator, perhaps, is that its output is insensitive to the input. No matter what input we feed in, the output is always the same—namely, true! This output insensitivity, perhaps, makes the smiley face operator less interesting and useful. Another way to see why the smiley face operator is unhelpful is to imagine what English phrase it might correspond to. What might it be a translation of? It might work pretty well as a translation of, 'It's either true or false that *p*.' Note that if '*p*' is false, then it's either true or false that *p*, and the same holds if '*p*' is true. If, however, we assume that truth and falsity are the only truth values, then '☺*p*', is just a boring triviality, true no matter what we plug in for '*p*.'

At any rate, there are lots of truth functions. Some, like the '→', will get included in Langerese. Some, like '☺', we can just ignore. The standard truth table for '→' looks like this:

Inputs		**Output**
p	q	$p \rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

This truth table displays a truth function that is often called material implication. Because '$p \rightarrow q$' is a complex statement containing two other statements, the truth table requires four lines. That's because the purpose of a truth table is to display all possible inputs to the truth function. Because there are two letters, each of which has two possible truth values, there are four possible combinations of truth values. In general, where *n* is the number of letters in a statement or statement form, the truth table will require 2_n lines.

It is extremely important to memorize the above truth table for the arrow. Let me say that again: It is extremely important to memorize the above truth table. It's going to turn out that the usefulness of Langerese really depends on memorizing a few basic things. I know that in our culture, rote memorization is widely seen as a bad thing, and teachers who ask students to memorize are sometimes seen as dampening students' creativity. This view is ridiculous. Of course, it's possible to overemphasize memorization. But memory is one of our basic cognitive faculties. Why shouldn't we try to develop and improve it? Often, creativity builds upon a foundation of memorization. Memorizing a repertoire of guitar licks does not make you a worse guitar player. Quite the opposite, it gives you the ability to combine musical phrases in new and creative ways. Something similar is true when it comes to learning a new language. The ultimate goal is to be able to express yourself creatively in the new language. But doing that requires countless hours of memorizing new vocabulary and grammatical rules. So please do not be deterred by the demands of memorization. Paradoxically, creativity requires a lot of it.

So stop. Memorize the truth table for conditional statements. Just memorize it. Don't think too hard about it. Once you have memorized it, resume reading. We need to think critically about that truth table, challenge it, and try to understand why it's constructed the way it is. But you should memorize it first.

Importantly, though, we now have our semantic theory in place for Langerese. We know how to give an interpretation of statement constants, and the truth table itself—the one you should be memorizing—is the interpretation of '→'.

Technical Terms

- Truth table
- Truth function
- Function
- Interpretation
- Material Implication

§8. The Bed of Procrustes

|Outline|

In Greek mythology, Procrustes was an ornery innkeeper. He insisted that every guest at his inn must fit into bed snugly. So if a guest was too short for the bed, Procrustes would stretch the poor person out, stretching-rack style. And if a guest was too tall, then things got even uglier, as Procrustes would lop off toes and even feet to guarantee a snug fit in bed. Our artificial language, Langerese, is like the bed of Procrustes, and English is like the guest at the inn. The basic problem is that the English term "if ... then ..." has nuances and connotations that are not reflected in the truth table for '→'. So some lopping and/or stretching may be required.

Take another look at the truth table for the '→' above. It has some strange features. One extremely odd feature is that there is no requirement that *p* and *q* have any relevance to each other. In order to make this vivid, consider two true statements that have nothing whatsoever to do with each other:

(4) Toby catches the frisbee.
(5) New Mexico shares a border with Colorado.

Obviously, both of these are true. But then the following conditional statement also comes out true, according to the truth table:

(6) If Toby catches the frisbee, then New Mexico shares a border with Colorado.

One tricky thing about (6) is that no one ever says things like that! We're not used to thinking much about conditional statements whose antecedents have nothing whatsoever to do with their consequents. You might even look at (6) and, using your own intuition, come to the conclusion that it's false. There are several related problems with (6). First, the antecedent and the consequent have almost nothing to do with each other. So there is a failure of **relevance**. Second, New Mexico and Colorado have shared a border for a long time. So in (6) the **temporal ordering** seems off. We'd expect the antecedent of a conditional statement to occur *before* the consequent—at least, if the conditional statement as a whole is true. Finally, and relatedly, there doesn't seem to be any **causal dependence** between the consequent and the antecedent. Obviously, Toby catching a frisbee is not the cause of New Mexico and Colorado sharing a border! With this in mind, consider the following principle:

Where it's true that 'if *p* then *q*,' the following things hold:

1. *p* is relevant to *q*;
2. the conditions described by *p* occur before those described by *q*;

3. the conditions described by *p* cause the conditions described by *q*.

These three things—relevance, temporal priority, and causation—are all baked into our ordinary, intuitive understanding of "if … then …," but the truth table for '→' does not capture these things at all. This is why I was careful above to say that '→' should not be taken to *mean* 'if … then…" Rather, the meaning of the arrow is given by the truth table.

To make this a bit clearer, consider a simple example, such as "If you flip the switch by the door, then the lights will go off." All three of these conditions hold in this case. Flipping the switch is relevant to the lights going off. Flipping the switch happens first. And it's also the cause of the lights going off. But if we translate this into Langerese, using (say) "$F \rightarrow L$," we lose a lot of the meaning. Once you realize this, you might wonder about the point of using the truth table to define '→'. Translating "if … then …" statements using the arrow involves significant loss, so what then is the gain? It turns out that keeping things truth functional really does have massive advantages, which will come into view very shortly.

Technical Terms

- Relevance
- Temporal ordering
- Causal dependence

§9. The Counter-intuitiveness of the Arrow

|Outline|

One serious pitfall of studying logic is relying too much on intuition, or on one's own feeling about whether or not something "makes sense." Of course we rely on intuition all the time in ordinary life. And sometimes, in logic, intuition leads in the right direction. If you just look at a *modus ponens* argument, you can sort of tell that the argument makes sense, and that the premises support the conclusion. But one goal of education should be to undermine our confidence in our own intuitions. Intellectual history, especially the history of science and mathematics, is just full of examples of things that seemed totally obvious to very smart people later on turning out to be completely wrong. For example, in mathematics, the idea that two parallel lines can never intersect has seemed to many of the most brilliant minds in history, from Aristotle on down to Kant, to be self-evidently true and indubitable. However, in the nineteenth century, the early developers of non-Euclidean geometry showed that you can drop that assumption and still have a viable system of geometry. In the natural sciences, very smart people have thought it was totally obvious that species cannot evolve into other species, or that vision works because the eye sends out rays that touch objects in the environment, or that the earth is at the center of the universe, or that human activities could not possibly alter the climate. The history of science is a graveyard of human intuition. We should certainly approach our own intuitions with a healthy dose of skepticism.

But having said all that, the truth table for the arrow is still very strange. Here it is again:

Inputs		**Output**
p	*q*	$p \rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

The last two, shaded lines, are especially odd. Those lines say that if '*p*' is false, then '$p \rightarrow q$' is automatically true, regardless of what '*q*' might be. In order to appreciate how odd this is, just think of any arbitrary false statement, such as:

(1) Socrates lived five centuries before Plato.

Any conditional statement having (1) as an antecedent will come out true! For example:

(2) If Socrates lived five centuries before Plato, then he was the teacher of Plato.
(3) If Socrates lived five centuries before Plato, then he invented the light bulb.

Intuitively, it's very hard to see how (2) could be true. If Socrates lived five centuries before Plato, it's hard to see how he could have been Plato's teacher in any but a metaphorical sense. (3) is also hard to make sense of. And yet in both cases, the false antecedent makes the entire conditional statement true.

Equally strange is the fact that a true consequent makes the entire conditional statement true. Take any arbitrary example of a true statement, such as:

(4) Socrates was Athenian.

Make (4) the consequent of a conditional statement, and that statement will be true, no matter what the antecedent is!

(5) If Socrates was born in San Francisco, then Socrates was Athenian.
(6) If Hartford is the capital of Connecticut, then Socrates was Athenian.

Both (5) and (6) come out true, just because the consequent is true. Intuitively, though, it's hard to see why (5) should be true. If Socrates was born in San Francisco, it's hard to see how he could be Athenian. And statement (6) involves an obvious failure of relevance. Hartford really is the capital of Connecticut, but why should that mean that Socrates was Athenian? In short, there are two facts about the '→' that are very strange: A false antecedent always guarantees that the conditional statement is true, and a true consequent always guarantees the same. In fact, the *only* case where the conditional statement comes out false is where the antecedent is true and the consequent is false. The counterintuitiveness of the truth table for the arrow is so severe that some people refer to these puzzles as the paradoxes of material implication.

Some philosophers and logicians think that these problems seriously limit the usefulness of the material conditional, or the '→'. Indeed, some think that instead of always translating 'If … then …' statements using the arrow, we should use some other logical operator that is not truth functional. That is controversial, but as you begin to study logic, it is really important to bear in mind that when you look at things and they simply do not add up, the problem might not lie with you. In this particular case, some very smart philosophers and logicians agree that the truth table for the arrow does not add up, and they have proposed various strategies for fixing things. Unfortunately, most of those strategies involve giving up on truth functionality.

Take the issue of relevance, which is a big one. Suppose we decide that in a true conditional statement, the antecedent must be relevant to the consequent. Well, in that case, we've already given up on truth functionality, and here's why. Suppose we have a conditional statement where both the antecedent and the consequent are true. So the truth values of the components are fixed. What can we say then about the truth value of the conditional statement as a whole? Nothing really, unless we look at what the antecedent and the consequent actually say—which is to say that we have to look at their *content* in order to see if the one is relevant to the other. Truth functionality has the advantage of being purely formal; you can't judge relevance without looking at content. And if we insisted upon a relevance requirement, then the truth values of the component statements—that is, of the antecedent and the consequent—will no longer uniquely determine the truth value of the whole conditional. Sometimes, conditional statements with a true antecedent and a true consequent will be true, whereas sometimes they will be false.

Perhaps the more obvious way to try to "fix" the counter-intuitiveness of the truth table for the arrow is just to revise the table itself. After all, since Langerese is our own invention, we can exercise our Godlike powers to set up truth tables however we want! So why not tweak the truth table a bit?

- Paradoxes of material implication

Technical Term

§10. Conjunction and Material Equivalence

|Outline|

Let's look again at the truth table for the arrow (although you should have it memorized by now!)

Inputs		Output
p	*q*	$p \rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

Since the last two lines are the really counterintuitive ones, maybe we could just change those 'T's to 'F's. Then we'd end up with the following:

Inputs		Output
p	*q*	$p \rightarrow q$
T	T	T
T	F	F
F	T	F
F	F	F

?

The truth values in the shaded boxes were changed. This quick fix might seem to avoid the failure of intuitiveness that we ran into in §9, but there's a problem. The problem is that there are *other* logical operators that we might want to have at our disposal in Langerese. Indeed, one of the most important logical operators in English and other natural languages is conjunction.

(1) Carlos is taking chemistry next semester, and he is also taking dance.
(2) *p* and *q*.

Statement (1) has the form expressed by (2). We could also translate (1) into Langerese, using a symbol for **conjunction**:

(3) $C \& D$

Just as we used a truth table to define the '→' operator, so must we use a truth table to define '&'. However, the truth table above is a natural one for conjunction. A conjunctive statement such as '*C* & *D*' is true just when '*C*' is true and '*D*' is true. But if either or both of those component statements, or conjuncts, is false, then the entire statement is false. Here, then, is the truth table definition for conjunction:

Inputs		Output
p	*q*	*p* & *q*
T	T	T
T	F	F
F	T	F
F	F	F

Of course, if we us this truth table to define conjunction, it makes no sense at all to use it to define conditionality as well! So this first proposed tweak won't work. Are there other possibilities?

One possibility is to change only the third line. If you do that, you get the following possible table for the arrow:

Inputs		Output
p	*q*	$p \rightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

?

The adjustment, once again, is indicated with shading. This table is problematic for much the same reason that the last one was. It's the table that's standardly used to define *another* logical operator, so it won't work for the arrow. This table says that '$p \rightarrow q$' is true whenever '*p*' and '*q*' have the same truth value. It's true when they are both true (on the first line) and true when they are both false (on the last line). If you think about it, this is not at all what 'If … then …" typically means. Rather this truth table serves as a good definition of a different operation, material equivalence, which is usually symbolized by '↔'. Sometimes you will also see material equivalence symbolized using the triple bar sign, '≡'.

Inputs		Output
p	*q*	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

In general, two statements are logically equivalent when they always have the same truth value: If one is true, then so is the other. And if one is false, then so is the other. Material equivalence (sometimes also called biconditionality, for reasons that will become apparent later) is any complex statement that asserts that its components have the same truth value. When '*p*' and '*q*' do have the same truth value, '$p \leftrightarrow q$' is true.

Remember that the original goal was to adjust the truth table for the arrow in an effort to reduce its counter-intuitiveness. But the number of possible truth functions is limited, and so far, all we've accomplished is to introduce truth table definitions for two new operators, '&' and '↔'! There's one more natural-seeming adjustment we could make to the original truth table for the arrow:

Inputs		Output
p	q	$p \rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	F

The change again is indicated with shading. Recall that when we changed both lines 3 and 4, we got the truth table for conjunction. When we changed line 3 only, we got the truth table for material equivalence. What if we change line 4 only? In that case, we'd get the very strange result that '$p \rightarrow q$' always has the same truth value as '*q*'! That just can't be the right result. If every conditional statement is logically equivalent to its own consequent, it's hard to see what the point of making conditional statements would ever be.

It seems that if we want to stick with the idea that the arrow should be a truth functional operator, then we're basically stuck with the standard truth table definition of it, as counterintuitive as it might be. As I said above, some people are more than willing to give up on truth-functionality. Perhaps there is a non-truth-functional operator that does a better job capturing our intuitions about "If … then …" On the other hand, I've also expressed my own doubts about the value of intuitions about these sorts of things. I want to invite you to remain open to the possibility that Langerese could be an incredibly powerful tool, and an intrinsically beautiful and elegant formal system, even if some very fundamental aspects of it fail to make intuitive sense.

Technical Terms

- Conjunction
- Conjuncts
- Material equivalence (or biconditionality)
- Logical equivalence

§11.Translating Conjunctions

|Outline|

In a natural language such as English, there are many ways of conjoining sentences:

(1) Hermione is taking philosophy next semester, and Ron is taking potions.
(2) Hermione is taking philosophy next semester; Ron is taking potions.
(3) Hermione is taking philosophy next semester, but Ron is taking potions.
(4) Although Hermione is taking philosophy next semester, Ron is taking potions.

In Langerese, all of these statements would be written the same way: '*G* & *H*'. Subtle differences in meaning and emphasis get left out. From a logical point of view, there's no difference between 'and' or 'but'.

In ordinary language, 'and' is sometimes used to mark temporal order. For example:

(5) Gretchen went on a vacation, and she got a job.

This way of putting things suggests that Gretchen went on the vacation *first*, and that she got a job *later*. This means something different from:

(6) Gretchen got a job, and she went on vacation.

Examples (5) and (6) show that sometimes when we say "and," we mean something like "and then." But this suggestion of temporal order gets left out entirely when we translate the statement into Langerese as '*V* & *J*'. In Langerese, we'll see, '*V* & *J*' is logically equivalent to '*J* & *V*'. The order doesn't matter.

Overall, translating conjunctions into Langerese is usually pretty easy. But even here, some meaning is often lost in translation.

§12. Non-truth Functional Operators

|Outline|

The arrow is a truth-functional operator. And for the time being, at least, we're constructing Langerese in such a way as to keep all the operators truth-functional. However, in order to understand anything really well, it helps a lot to see what the contrast class looks like. In this case, what would a non-truth functional operator look like?

A good example of a non-truth functional operator is necessity. Necessity is what's known as a **modal operator**. 'Modality' just means anything having to do with possibility and necessity. Modal logic is the study of systems of logic that make use of these notions. Before proceeding, it's worth noting that the concept of necessity has long been a point of contention among philosophers. For example, many philosophers have wanted to establish that God is a **necessary being**, or a being that cannot not exist—a being whose non-existence is impossible. A **contingent being**, by contrast, is one that might not have existed. For example, you and I are contingent beings. Many philosophers have been interested in the notion of **necessary truth**. Are there any statements that must be true, statements that cannot possibly be false? A contingent truth is a statement that might have been false. Other philosophers, such as David Hume, have expressed skeptical doubts about the whole idea of necessity. If, as Hume and other empiricists have held, all of our concepts are derived from sense experience, how could we have a concept of necessity at all? Where would such a concept come from? Our five senses do not give us any access to possibility and necessity. In the twentieth century, W.V.O. Quine was also skeptical about modal notions, partly on the grounds that modal operators are not truth functional, and truth functional logic is all we need for most purposes.

Traditionally, logicians have used the box symbol, '□', to stand for the necessity operator. So you'd read '□*p*' as 'Necessarily *p*.' What would a truth table for this new operator look like? It's not clear at all:

Input	Output
p	□*p*
T	?
F	F

It seems fairly clear that if 'p' is false, then '□*p*' should also be false. That's because '□*p*' means something like 'It just has to be true that *p*.' But if '*p*' is false, then it's not the case that it has to be true that *p*. The problem is with the first line. If '*p*' is true, that says nothing whatsoever about whether '*p*' *must* be true. For this reason, modal operators are not truth functional.

Incidentally, some philosophers and logicians like modal operators because they give us a new way of expressing conditional statements. We saw above that there are some issues with material implication; to handle those issues, the logician C.I. Lewis argued that we should instead translate conditional statements using **strict implication**:

$$\Box(p \to q)$$

This idea has some promise, but it's also crucial to bear in mind that any time we add a new operator to Langerese, we need a semantic theory for that operator. As long as our operators are truth functional, the semantic theory is easy peasy. However, modal operators require their own semantic theory. The standard approach is known as **possible worlds semantics**. You stipulate that '$\Box p$' is true just in case 'p' is true in all possible worlds, and go from there. Modal operators and the associated semantic theory raise lots of philosophical and logical questions, questions that we can safely set aside for another occasion. (One of the biggest questions is just: What in the heck is a possible world?) Most beginning logic textbooks scarcely mention these issues, but I think it's important to see just one example of a non-truth functional operator, and to get a glimpse of some of the problems we'd have to address if we used such operators in Langerese.

For now, I would propose that we try to stick with truth-functional operators, just to get an appreciation for how much we can do with how little. We will always have the option of adding modal operators to Langerese later on if we decide we need them. One thing that I especially love about logic is the *spirit of minimalism*. As we develop our artificial language, the idea is to try to get the most from the least. How simple can we keep our artificial language, and still have a useful tool? (A tool for doing what, you might ask? We are just about to get to that.)

Technical Terms

- Modal operator
- Strict implication
- Necessary truth
- Necessary being
- Contingent being
- Possible worlds semantics

§13. Negation and Disjunction

|Outline|

So far, Langerese has three logical operators: '→', '↔', and '&'. In other words: the arrow, the double arrow, and the ampersand. Traditionally, most logicians have wanted to add at least two more operators to the mix: '~' and '∨', for the operations of negation and disjunction. **Negation** is really easy. The tilde, or '~', just corresponds roughly to 'not'.

Input	Output
p	$\sim p$
T	F
F	T

Although it might look like statements having the form '~ *p*' are simple, in fact they are complex or compound. That's because every negation contains at least one other simple statement that can stand on its own.

So where '*p*' is true, '~ *p*' is false, and vice versa. In Langerese, we can also use the wedge '∨' as a translation of 'or'. We can translate '*p* or *q*' as '*p* ∨ *q*.' The conventional truth table for the wedge is as follows:

Inputs		Output
p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

The technical term for this truth function is **disjunction**, and '*p*' and '*q*' are both **disjuncts**.

Sometimes, when speaking colloquially, we use 'or' in an exclusive way. For example, suppose that Hermione says, "Either I'm taking physics next semester, or else I'm taking chemistry." One might reasonably take her to mean that she's not taking both physics and chemistry. Context matters a lot in a case like this. Suppose we have a bit of background information: At Hogwarts (let's say), students are required to take at least one science course, and many students choose to do the bare minimum. In that context, it makes sense to treat Hermione as making an **exclusive disjunction**. In other contexts, though, we use disjunction in an inclusive way, to mean something more like "and/or." For example, suppose you are having some friends over. They offer to bring some munchies along, and they ask you for suggestions. You might say, “Bring chips or cookies.” Of course, if your friends showed up with both chips and cookies, you probably would not be too upset, and would not view this as a violation of your instructions. If they showed up with chips only, that would be fine too. In this case, the “or” in “chips or cookies” is an **inclusive disjunction**. What you really mean to say is “bring chips and/or cookies.”

Technical Terms

- Negation
- Disjunction
- Disjuncts
- Exclusive disjunction
- Inclusive disjunction

§14. Unless

|Outline|

Maybe it's just me, but I really hate the word 'unless'. For some reason, I find myself able to think somewhat clearly about issues in logic unless the topic is 'unless.' It's a good word to think about, though, because we use it all the time, and it's not always clear how to translate it into the symbolism of Langerese. This, in other words, is a lovely example of the Bed of Procrustes. We only have a handful of logical operations: negation, material implication, conjunction, disjunction, material equivalence. 'Unless' is not one of these. But unless we just resolve to stop using the word, we need some way of translating it into Langerese.

(1) Harry has his wand unless Draco took it.

How should we translate this statement? 'Unless' looks a lot like a logical connective, but it doesn't obviously correspond to any of the operators of Langerese. In a case like this, the best approach is to do a little brainstorming, come up with different possible translations, and then weigh their pros and cons.

To start with the easy part, let's designate some sentence letters:

H: Harry has is wand.
D: Draco took it.

One possible translation that seems to capture the meaning of the original is:

(2) $\sim H \rightarrow D$

If Harry doesn't have his want, then Draco took it. This is a good translation of (1), but other ideas might occur to you as well. How about:

(3) $\sim D \rightarrow H$

If Draco did not take Harry's wand, then Harry has it. Or how about this:

(4) $H \vee D$

Either Harry has his wand, or Draco took it. Now it so happens that all of these possible translations—(2), (3), and (4)—are logically equivalent. It will take some additional work to prove this, but if you're curious about how to do it, please skip ahead to §25. Logically equivalent statements always have the same truth value. This means, in a way that they are all interchangeable with each other. If you are making an argument, and you exchange (2) for (3), that swapping of one statement for its logical equivalent will make no difference to anything.

This, in turn, raises an interesting issue with translation from English into Langerese. In general, when you're faced with a choice like that between (2) and (3), where you can translate an English statement into either of two different Langerese statements that are logically equivalent, you can go either way. There is, in effect, no wrong answer, and the decision is just a matter of taste. Actually, let me be really clear: there are *some* bad translations of statement (1) above—some wrong answers. But neither (2) nor (3) nor (4) is wrong.

We may be getting a bit ahead of ourselves, but we'll see later on that every statement in Langerese is logically equivalent to indefinitely many other statements. The easiest way to see this is by reflecting on double negation. ‘p’ is always logically equivalent to ‘$\sim\sim p$’. Thus, “I have a dog” is equivalent to “It's not the case that it's not the case that I have a dog.” This means that it's never wrong, when translating into Langerese, to tack two tildes out in front of the statement!

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

This is also a perfectly good translation of (1), and it's logically equivalent to (4). But what we just said about adding on two tildes applies to (5) as well!

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

And you could keep on doing this for as long as you wish! Yikes. In practice, you should definitely avoid translations such as (5) and (6), just because their complexity makes them more cumbersome to work with. In general, where you have two logically equivalent statements, one of which is longer than the other, it is usually preferable, on aesthetic and pragmatic grounds, to go with the shorter option.

§15. Necessary *vs.* Sufficient Conditions

|Outline|

The distinction between necessary and sufficient conditions is an especially helpful tool of clear thinking. Indeed, one traditional goal of philosophy has always been conceptual clarity, and many philosophers have tried to achieve clarity by identifying necessary and sufficient conditions for the correct application of a concept. This approach is sometimes called **conceptual analysis**. Although it's somewhat controversial as a philosophical method, it's really important to get a feel for how conceptual analysis works, if only so that you can better appreciate the methods that some other philosophers employ.

A **necessary condition** is basically a prerequisite. To say that one thing is a necessary condition for a second thing means that if you don't have the first thing, you can't have the second thing. For example:

Being an animal is a necessary condition for being a cat.

Being a polygon is a necessary condition for being a triangle.

Passing the bar exam is a necessary condition for practicing law.

Hopefully the basic idea here is easy to see. You could also say that passing the bar is a prerequisite for practicing law. If you don't pass the bar exam, you cannot practice law—at least not legally.

Interestingly, a **sufficient condition** is like the mirror image of a necessary condition. Intuitively, it might help to think of a sufficient condition as a guarantee. To say that *A* is a sufficient condition for *B* is to say that having *A* guarantees that you have *B*. For example:

Being a cat is a sufficient condition for being an animal.

If you are a cat, then that guarantees that you are an animal. Notice that this is just the reverse of the example above. In general, if *A* is a necessary condition for *B*, then *B* is a sufficient condition for *A*.

Being a triangle is a sufficient condition for being a polygon.

Practicing law is a sufficient condition for having passed the bar exam.

This last example is just a bit trickier than the others. The claim is not about cause and effect. It would be silly for an aspiring attorney to think, "Oh I don't need to study for the bar exam. If I just start practicing law, then that guarantees I will pass the bar." The thought, rather, is that if you met someone who is—legally!—practicing law, then you could infer that she has passed the bar. If she is practicing law, then that guarantees that she must have passed the bar exam already.

Technical Terms

- Conceptual analysis
- Necessary conditions
- Sufficient conditions

§16. Translating Conditional and Biconditional Statements

|Outline|

One of the most difficult things about learning logic is getting comfortable with standard ways of translating conditional statements. It is a good idea to work hard at memorizing the material in this section. This is one of many cases in logic where, if you just try to wing it and think things through as you go, it is very easy to make mistakes.

The following is a list of different ways of saying exactly the same thing:

$P \rightarrow Q$
If P, then Q
P is a sufficient condition for Q
P implies Q
P guarantees that Q
P only if Q
Q if P
Q is a necessary condition for P
Q is a prerequisite for P

If this is hard to wrap your mind around, it might help to stick with simple examples:

If Garfield is a cat, then Garfield is an animal.

Garfield's being a cat is a sufficient condition for Garfield's being an animal.

Garfield's being a cat implies that Garfield is an animal.

Garfield's being a cat guarantees that Garfield is an animal.

Garfield is a cat only if Garfield is an animal.

Garfield is an animal if Garfield is a cat.

Garfield's being an animal is a necessary condition for his being a cat.

Garfield's being an animal is a prerequisite for his being a cat.

Every single one of these statements is true, and they all say exactly the same thing! Notice how this is related to the point made in the previous section—namely, that necessary and sufficient conditions are like mirror images of each other.

Another possibly helpful way to think about this is to consider what you might call the direction of reading for a conditional statement. Take a simple conditional statement:

$P \to Q$

If you read this in the usual way, *from left to right*, you get logical sufficiency, and you'd also read it as an "only if" statement. But if you read it backwards, *from right to left*, you get logical necessity, and you'd read it as an "if" statement.

One extremely important thing to bear in mind is that the following two statements are *not* equivalent to one another:

$P \to Q$
$Q \to P$

These say totally different things. For example:

(1) If Toby is a dog, then Toby likes to sit on the sofa.

(2) If Toby likes to sit on the sofa, then Toby is a dog.

The first statement (1) is at least sort of plausible. Don't all dogs like to sit on the sofa? But statement (2) is obviously false. Toby might be a cat, or a human, and still enjoy sitting on the sofa. It's possible for (1) and (2) to have different truth values, which means that they are not logically equivalent. One consequence of this is that when translating conditional statements, it matters a great deal what order you put things in!

We saw in §10 above that biconditional statements, or statements using the triple bar, have their own distinct truth table. They are actually a little bit easier to translate:

$P \leftrightarrow Q$
P if ***and*** only if Q
P is both necessary ***and*** sufficient for Q
P implies Q, ***and*** Q implies P

A biconditional claim basically combines two conditional statements into one.

$$\underbrace{\begin{array}{lll} P \text{ only if } Q & \qquad\qquad & P \text{ if } Q \\ P \to Q & & Q \to P \end{array}}$$

P if and only if Q
$P \leftrightarrow Q$

Now it's a little easier to see why we use the double arrow '↔' for the biconditional. Think of a biconditional as a two-way street, where the statement on each side implies that on the other. A regular conditional statement, by contrast, is a one-way street.

Translation from English into an artificial language like Langerese is always tricky. The only way to get good at it is to practice a lot, and you are bound to run into unforeseen glitches and challenges. But if you start out by mastering this standard approach to translating conditionals, then everything else will be a lot easier.

§17. Fun with Longer Statements

|Outline|

Here is an example of a really long statement in Langerese:

$(S \rightarrow A)$ & $\{P \leftrightarrow [(S \vee D)$ & $(D \rightarrow P)]\}$

When you first look at a statement like this, it might seem like a lot of gibberish. But it is actually a wff! With just a little bit of patience, it's easy to get comfortable working with statements like this, and even much longer ones.

Before going any further, note that the above statement includes both parentheses and brackets. The different bracket symbols are technically all the same. Using different ones just makes the thing a little easier to read. The technically correct way to write it is:

$(S \rightarrow A)$ & $(P \leftrightarrow ((S \vee D)$ & $(D \rightarrow P)))$

The only problem with writing it this way is that you have to be careful to count the parentheses to make sure you have the right number.

Here is one perfectly good way to read the above statement. Let's call it the **formalist reading**:

> "Open bracket S arrow A close bracket ampersand open bracket P double arrow open bracket open bracket S wedge D close bracket ampersand open bracket D arrow P close bracket close bracket close bracket."

This formalist reading is kind of fun when you get used to it. This is a purely syntactical reading. When you read the statement this way, you're just listing a string of symbols without any care or concern for what they might mean or stand for. The formalist reading, in other words, just ignores issues of interpretation.

Now suppose we try to interpret the statement. It's really up to us how to do that. We have to assign sentences to each letter (or each sentence constant), but we can assign any English sentences we wish. For example:

S: Socrates drinks hemlock
A: Aristotle drinks hemlock
P: Plato drinks hemlock
D: Diogenes drinks hemlock

With these assignments in place, we can actually transform that statement in Langerese into something in English.

$(S \rightarrow A)$ & $\{P \leftrightarrow [(S \vee D)$ & $(D \rightarrow P)]\}$

> If Socrates drinks hemlock then Aristotle drinks hemlock, and Plato drinks hemlock if and only if either Socrates or Diogenes drinks hemlock, and Diogenes drinking hemlock guarantees that Plato drinks hemlock.

The trick to doing this is to try to break the statement into chunks. Each chunk gets translated as its own unit. The ampersand in the middle of the statement is the main operator, because it binds together the two largest chunks—the yellow chunk and the green chunk. Notice also that the longer statement as a whole is a conjunction, having the form

$$p \,\&\, q$$

> yellow chunk & green chunk

Now focus on the green chunk, taking it all by itself.

$$\{P \leftrightarrow [(S \lor D) \,\&\, (D \rightarrow P)]\}$$

Remember that our earlier convention, rule [S4] from §6, which says that for convenience we're allowed to drop the brackets on the outside of a statement. Since the purpose of the brackets is to help us keep track of the main operator, we don't really need the ones on the outside. So:

$$P \leftrightarrow [(S \lor D) \,\&\, (D \rightarrow P)]$$

Here the main operator is the double arrow. We can divide this statement into chunks of its own, in turn:

$$P \leftrightarrow [(S \lor D) \,\&\, (D \rightarrow P)]$$

Referring back to our English interpretation of this statement, we can take it to say:

> Plato drinks hemlock if and only if either Socrates or Diogenes drinks hemlock, and Diogenes drinking hemlock guarantees that Plato drinks hemlock.

The procedure we're using here is really simple: just find the main operator of a statement. When you do, you'll see that the main operator connects two different chunks. Look at each chunk on its own. If it has any operators, find the main one. That one will also connect two different chunks. Just keep going until you've decomposed the statement into its individual letters.

Translating statement from English into Langerese works in much the same way, except that you're going in the opposite direction. Let's start with a long and cumbersome statement in English:

> Either Harry goes to the banquet or Snape does not go, and Hermione goes to the banquet if and only if Ron’s going to the banquet implies that Professor McGonigall goes, too.

Of course, nobody talks this way. This is the sort of statement that you only find in logic puzzles. But we certainly could talk this way if we wanted to. For the moment, though, the important thing is just to see how to translate a longer statement like this one. If you can do this, then other, more realistic translations will be much easier.

First, we need some sentence letters. Try to find every simple statement within this larger statement. Which are the simple statements that cannot be broken down any further, because they contain no logical operators?

H: Harry goes to the banquet
S: Snape goes to the banquet
G: Hermione goes to the banquet
R: Ron goes to the banquet
M: Prof. McGonigall goes to the banquet

It makes no difference at all which letters you use. I used ‘G’ for Hermione because her last name is Granger. It’s generally a good idea to choose letters that help you remember which letter goes with which statement. But that isn’t essential. We could have just used *A, B, C, D* …

You might be wondering why we don’t say:

S: Snape does not go to the banquet

The issue here is that “not” is a logical operator. We actually have a way of translating it using logical symbolism, ‘~ *p*’. For this reason, ‘Snape does not go to the banquet’ is technically a complex or compound statement (§ 1). It’s a statement with logical structure. As a general rule, when translating, you want to show all the logical structure you can; the idea is to put the logical structure on display. So for that reason, sentence letters should only stand for simple statements. If ‘*S*’ stands for ‘Snape goes to the banquet,’ then we can translate ‘Snape does not go to the banquet’ as ‘~ *S*.’

Once we’ve assigned sentential letters to each simple component, the next step is to identify the main operator of the longer statement. Here’s the statement again:

> Either Harry goes to the banquet or Snape does not go, and Hermione goes to the banquet if and only if Ron’s going to the banquet implies that Professor McGonigall goes too.

Here the main operator is ‘and’, and the whole statement is a conjunction. It can sometimes be a little challenging to find the main operator, and there is no perfect method for doing so. In this case, the grammar of the statement helps a bit. Note the location of the comma: that bit of punctuation divides the sentence nicely into two chunks which are connected by the ‘and’. So when you look at the logical structure of the statement as a whole, it looks like this:

_______ and ______________________

Translation is easiest if you approach it one chunk at a time. So consider the chunk on the left: “Either Harry goes to the banquet or Snape does not go.” That’s actually pretty easy to translate:

$H \vee \sim S$

If we fill this in, the statement begins to take shape:

$(H \vee \sim S)$ and ______________________

Note that we have to add brackets here in order to mark the boundary of the left-hand chunk.

Now look at the right hand chunk:

> Hermione goes to the banquet if and only if Ron’s going to the banquet implies that Professor McGonigall goes too

The main operator of the right-hand chunk (highlighted) is a triple bar. Once you see that, the right-hand chunk is also pretty easy to translate:

$G \leftrightarrow (R \rightarrow M)$

Now we can fill that in to the larger statement:

$(H \vee \sim S)$ and $[G \leftrightarrow (R \rightarrow M)]$

Note once again that we had to add brackets around the right-hand chunk, so as to mark its boundaries. Now let’s get rid of the English word ‘and’, which is contaminating the nice formula here, and replace it with symbolism:

$(H \vee \sim S)\ \&\ [G \leftrightarrow (R \rightarrow M)]$

And we now have an elegant translation of the long, ugly sentence:

> Either Harry goes to the banquet or Snape does not go, and Hermione goes to the banquet if and only if Ron’s going to the banquet implies that Professor McGonigall goes too.

This is all there is to translation.

It can sometimes be really helpful to think of translation between a natural language (such as English) and an artificial language (like Langerese) as analogous to more familiar kinds of translation, like Arabic-English translation. In some ways, though, this analogy is not perfect, because statements in any natural language, whether English, or Arabic, or Japanese, or whatever, have both form and content. Statements in our artificial language, Langerese, are purely formal. They are what's left when all the content and meaning has been drained out. So the process of translating from English to Langerese could be described as a process of abstraction.

To make this last point really vivid, note that there are indefinitely many ways to interpret the formal statement under consideration:

$$(H \vee {\sim}S)\ \&\ [G \leftrightarrow (R \rightarrow M)]$$

For example:

> Either hadrosaurs are extinct or spinosaurs are not extinct, and hadrosaurs are extinct if and only if rhamphorhynchus being extinct implies that mosasaurs are extinct.

This has the same form as the longer statement about the Harry Potter characters, but different content. When we think and reason in everyday life, we often tend to fixate on the content, or on what it is that we're thinking about. When doing logic, however, the goal is to let the content fall away, and let the form shine through.

- Formalist reading

Technical Term

§18. Calculating Truth Values for Statements

|Outline|

Have you memorized the truth table definitions for all of the logical operators? If not, read no further. Stop what you are doing. Go back and reread §7, §10, and §13. You must memorize the truth table definitions for '~', '→', '&', '∨', and '↔'. If you haven't done so, then from here on out, you are really wasting your time. I know that culturally speaking, many people look down on rote memorization as an approach to learning that is somehow stultifying and soul-crushing. And sure, rote memorization can be soul-crushing if it's over-emphasized. But sometimes there is just no other way. Sometimes it can also feel good to flex your memory and see what it can do. If you don't memorize the truth tables, you will likely find logic to be confusing and depressing. If you do memorize them correctly, you will find much of what follows to be really easy.

One hugely important aspect of sentential logic is the calculation of truth values. Once you've memorized the truth table definitions for the operators, this turns out to be incredibly easy to do. It can also be somewhat relaxing. Just think of yourself as a mindless computer that takes certain inputs and generates certain outputs.

Consider the following statement:

$(P \rightarrow Q) \leftrightarrow ({\sim} R \rightarrow Q)$

Earlier we saw that sentential logic is truth functional, which just means that the truth value of any complex statement (such as this one) is uniquely determined by the truth values of the simple statements it contains. The basic idea of truth functionality is really simple: If you know what truth values to attach to the letters in the statement above, then you can easily figure out what the truth value of the whole statement must be.

In order to see how this works, let's just stipulate:

P: true
Q: true
R: false

What would the truth-value of the whole statement be? The statement as a whole is a biconditional, since the main operator is a triple bar.

________ ↔ ________

If we knew whether the chunk on each side is true or false, then we could consult the truth table definition for the triple bar to see what the whole statement must be. (And that is why it's so important to memorize the truth table definitions for the logical operators.) So let's take each chunk on its own. The chunk on the left is:

$$P\rightarrow Q$$

We’ve stipulated that ‘P’ and ‘Q’ are both true, so ‘$P\rightarrow Q$’ must be true. So the left-hand chunk is true.

$$\begin{array}{ccc} \underline{(P\rightarrow Q)} & \leftrightarrow & \underline{({\sim}R\rightarrow Q)} \\ \text{T} & & \end{array}$$

The right-hand chunk is just a bit more complicated. We know that ‘Q’ is true. We’ve also stipulated that ‘R’ is false. But that means that ‘${\sim}R$’ is true. Since both ‘${\sim}R$’ and ‘Q’ are true, it follows that ‘${\sim}R\rightarrow Q$’ is also true. So the right-hand chunk is true as well.

$$\begin{array}{ccc} \underline{(P\rightarrow Q)} & \leftrightarrow & \underline{({\sim}R\rightarrow Q)} \\ \text{T} & & \text{T} \end{array}$$

Now recall that a biconditional statement basically asserts that the chunk on the left and the chunk on the right have the same truth value. Here that is indeed the case: the left-hand chunk and the right-hand chunk are both true. So the whole statement is true. We can indicate this by placing a ‘T’ under the main operator.

$$\begin{array}{ccc} \underline{(P\rightarrow Q)} & \leftrightarrow & \underline{({\sim}R\rightarrow Q)} \\ \text{T} & \text{T} & \text{T} \end{array}$$

One reason why it’s important to practice these truth value calculations is that this is the key to filling in truth tables for propositions. Suppose we wanted to do a full blown truth table for the above statement. It would start out looking like this:

P	Q	R	${\sim}R$	$P\rightarrow Q$	${\sim}R\rightarrow Q$	$(P\rightarrow Q)\leftrightarrow({\sim}R\rightarrow Q)$
T	T	T				
T	T	F				
T	F	T				
T	F	F				
F	T	T				
F	T	F				
F	F	T				
F	F	F				

First, a word about the set-up. Earlier when we were looking at the truth table definitions for the operators, the truth tables were all really short. Remember that the purpose of a truth table is to display all possible combinations of truth values. If you have two letters, then that means there are four possibilities (2 x 2). If you have three letters, there are eight possible combinations of T’s and F’s (2 x 2 x 2). And so on. The general rule is that where n is the number of sentential letters, your truth table will have to have 2_n lines.

When we started thinking about truth value calculation, we just stipulated that P and Q were true, while R is false. (Of course we can make any stipulation we want.) Then the task was to calculate the truth value of the whole statement based on that starting assumption. Notice, however, that our opening stipulation just amounted to zeroing in on one line of this full truth table:

P	Q	R	$\sim R$	$P \rightarrow Q$	$\sim R \rightarrow Q$	$(P \rightarrow Q) \leftrightarrow (\sim R \rightarrow Q)$
T	T	T				
T	T	F				
T	F	T				
T	F	F				
F	T	T				
F	T	F				
F	F	T				
F	F	F				

The line highlighted in green is the one we were effectively considering, where we assume that P and Q are true, while R is false. And what we just figured out was that the whole statement in the right-hand column comes out true on the green line. As an exercise, you might go ahead and fill in the rest of the truth table on your own.

A couple of further details about truth tables: first, note how the truth table above is set up. With three letters, you begin by alternating between four T's and four F's. Then you alternate by twos: two T's, two F's, two T's, etc. as you read down the column under Q. Finally, you alternate by ones: T, F, T, F, down the column. This set-up guarantees that you will have every possible combination of truth values for P, Q, and R represented in the table. One question that comes up occasionally is what would happen if you changed the order. For example, what would happen if under the P column you alternated the T's and F's by twos, while under the Q column you alternated by fours. The honest answer is that it makes no difference, *as long as* you are careful to set your table up so that all possible combinations of truth values are represented. One consideration, though, is that if you organize your table as above, whereas I organize mine as below, then our tables will look different, even though both are correct and ultimately give the same answers to any questions we might care about.

P	Q	R	$\sim R$	$P \rightarrow Q$	$\sim R \rightarrow Q$	$(P \rightarrow Q) \leftrightarrow (\sim R \rightarrow Q)$
T	T	T				
T	T	F				
F	T	T				
F	T	F				
T	F	T				
T	F	F				
F	F	T				
F	F	F				

This table is perfectly fine, but it will yield a different pattern of T's and F's than the one above. It will be much easier for us to compare our truth tables if we all stick by the rule of working inward: alternating by fours in the first column, by twos in the second, and so forth. This "rule" isn't necessitated by anything in logic, but it will make things more convenient for everybody.

§19. Tautologies

|Outline|

Some statements in Langerese have a very special status. When you look at them, you see that they just have to be true:

$P \vee {\sim}P$
$P \rightarrow P$
${\sim}(P \,\&\, {\sim}P)$

If you say, “Either I’ll take chemistry next semester or I won’t,” that is obviously true. In a way, it doesn’t make any difference what I actually choose to do. This statement, which has the form ‘$P \vee {\sim}P$ comes out true no matter what. This is an example of a **Tautology**. A tautology is a complex statement that comes out true no matter what truth values you assign to its simplest components. Some people have also referred to such statements as laws of logic. In fact, this particular example, ‘$P \vee {\sim}P$ is known as the law of excluded middle. Some people refer to these statements as theorems of the logical language—we might say theorems of Langerese. Others call them valid sentences. For now, we can just call them tautologies.

In order to see what makes tautologies so special, it helps to construct a truth table. There are two different ways to construct a truth table for a single statement. Here is one way:

P	${\sim}P$	$P \vee {\sim}P$
T	F	T
F	T	T

Using this approach, the statement you care about—in this case ‘$P \vee {\sim}P$—ends up in the column on the far right. The truth table only has two lines, because there is only one letter—P—and P can have either of two truth values. Obviously, if P is true, then ${\sim}$P has to be false, and vice versa. In general, when filling in a truth table like this, the truth table definitions for each logical operator will tell you how to do it. So the truth table for negation ‘${\sim}$’ tells you how to fill in the column under ‘${\sim}P$. Similarly, the truth table definition for ‘$\vee$’ tells you how to fill in the column under ‘$P \vee {\sim}P$.’ According to the truth table for the ‘$\vee$’, the whole disjunctive statement is true if either disjunct is true. On the truth table above, P is true on the first line, which makes ‘$P \vee {\sim}P$ true. But then ‘${\sim}P$ is true on the second line, which also makes ‘$P \vee {\sim}P$ true! So the statement on the right is a tautology.

In general, when you fill in a truth table for a statement and the statement comes out true on every line, that means it's a tautology. One could even define a tautology as a statement that comes out true on every line of a truth table, or a statement that comes out true no matter how you assign truth values to the sentence letters it contains. You might also think of a tautology as a statement that is *true in virtue of its form*. You could think of it as a statement that's true no matter how we interpret it—no matter what we suppose the letters to stand for. Some logicians might thus define a tautology as a statement that's *true under every interpretation*. Think of each line on the truth table as representing one possible interpretation of the statement—one way of assigning truth values to the sentence letters. In that case, saying that a tautology is true under every interpretation is another way of saying that it's true on every line of the truth table.

There are indefinitely many tautologies in Langerese. Here's why: Suppose that p is a tautology. In that case you can form a new tautology by adding two negation signs: $\sim\sim p$. ‘$\sim\sim p$’ will always have the same truth value as ‘p’. But then why not add two more negation signs, just for fun? You'd get ‘$\sim\sim\sim\sim p$’, which would still be a tautology. Of course, you can keep adding pairs of negation signs *ad infinitum* without ever violating the grammatical rules of Langerese, so there would seem to be infinitely many tautologies.

Simpler tautologies, like the ones above, are really easy to see and appreciate. You probably don't really need a truth table in order to tell that ‘$P \vee \sim P$ has to be true. However, in logic you can quickly get into trouble if you rely too much on your own intuitive grasp of things. The problem is that some tautologies are considerably longer—some even vastly longer. And even ones that are just a bit longer might not look immediately like tautologies. Here is one example:

$$P \to (Q \to P)$$

Is that really a tautology? It doesn't seem obvious. (If you can look at that and tell intuitively that it's a tautology, then congratulations. Your intuition is maybe a tiny bit more reliable than most people's, but the thing is: as statements get more complex in logic, nobody's intuition is worth very much.) Still you can prove that this is a tautology by filling in a truth table:

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

You should be able to fill the rest in on your own at this point. One quick tip, though: when filling in the column under ‘$Q \to P$, note that Q is the antecedent of the conditional. So on the second line, for instance, you have a conditional with a false antecedent and a true consequent.

Truth tables are a good **decision procedure** for tautologies. In general, a decision procedure is just a test that you use to figure out whether something has a given property. For example, at litmus test is a decision procedure for determining whether something is an acid. If you have some liquid in your test tube and you want to know if it's acidic, just stick a strip of litmus paper in the liquid and see if it changes color. Truth tables give you a kind of litmus test for tautologies. If you want to know whether a statement is a tautology, just construct a full truth table for it. If the statement comes out true on every line, then it is indeed tautologous.

Technical Terms

- Tautology
- Law of excluded middle
- Decision procedure

§20. Two Ways of Constructing Truth Tables

|Outline|

It's worth mentioning that there are a couple of different ways of designing truth tables. It makes no difference at all which approach you use (though it might be good to pick one approach and stick with it.) Here again is the set-up for a truth table:

P	Q	R	$\sim R$	$P \rightarrow Q$	$\sim R \rightarrow Q$	$(P \rightarrow Q) \leftrightarrow (\sim R \rightarrow Q)$
T	T	T				
T	T	F				
F	T	T				
F	T	F				
T	F	T				
T	F	F				
F	F	T				
F	F	F				

Notice that with this approach, each sentence letter gets a column on the left. Then each chunk of the statement gets its own column. Finally, the target statement gets its own column on the right. The T's and F's in the right-hand column will be the truth values for the target statement.

A second way to construct a truth table is just to write out the target statement at the top and then place everything in columns underneath it:

$(P$	$\rightarrow$	$Q)$	$\leftrightarrow$	$(\sim$	R	$\rightarrow$	$Q)$
T		T			T		T
T		T			F		T
T		F			T		F
T		F			F		F
F		T			T		T
F		T			F		T
F		F			T		F
F		F			F		F

This is exactly the same truth table—and will yield exactly the same answers—only set up in a different fashion. The green column under the double arrow—the main operator—represents the truth value of the whole statement. As before, you start out by filling in the T's and F's under the individual sentence letters. Notice, though, that because *Q* appears twice, you have to fill in its column twice. Make sure that the alignment of T's and F's is exactly the same in both *Q* columns. From here, you just proceed to fill in the table based on the truth table definitions of the operators.

(*P*	→	*Q*)	↔	(~	*R*	→	*Q*)
T	T	T			T		T
T	T	T			F		T
T	F	F			T		F
T	F	F			F		F
F	T	T			T		T
F	T	T			F		T
F	T	F			T		F
F	T	F			F		F

Here as a next step I've filled in the brown column under the arrow, again using the truth table definition for the arrow as a guide. To complete the table, just work towards the center (green column), going chunk by chunk. Because the tilde '~' operates on '*R*' alone, be sure to do that column before doing the column under the second arrow.

§21. Human Limitations

|Outline|

Truth tables are relatively fun and easy to complete for short statements. But as statements get longer, the truth tables get incredibly long and annoying. In principle, the truth table technique is a perfectly reliable decision procedure for tautologies. If a statement is a tautology, the truth table will say so. And if it's not, the truth table will also say so. There will never be any false positives or false negatives. However, the size of the truth table grows exponentially as you add letters. A statement with 4 letters will have 16 lines. 5 letters means 32 lines. 6 letters means 64 lines.

$$[A \rightarrow (B \vee C)] \rightarrow [\sim D \,\&\, (E \vee \sim F)]$$

That is actually not such a long statement. But with six letters, it would take you all afternoon to fill in the complete 64-line truth table! And by the end of it, you would be cursing the fates and wondering why you ever decided to study logic. Every once in a while, I can persuade myself to do a 16-line truth table, but after that, I'm out. The problem here is not just a matter of getting bored and cross-eyed, either. Due to ordinary human limitations (limited attention, bad handwriting, etc.) the probability of making a mistake actually increases the longer the truth table. Even once you get good at it, there is always some nonzero probability of filling in a T where there should be an F. (I do that occasionally too after many years of practice.) The more T's and F's you have to fill in, the greater the probability that there will be a mistake somewhere in the truth table.

One question that we need to explore is whether there are any other decision procedures that might be easier to use than what we might call the long truth table method. Luckily, there is. It's possible to save yourself a lot of pain and annoyance by working backwards. Later on, we'll see that there is another rather different (and more fun) way to test for tautologies: we can try to prove them. But the backwards truth table method—sometimes also called the indirect or abbreviated method—is one that we can put to work right away.

§22. Short Truth Tables for Tautologies

|Outline|

Consider once again that long statement whose truth table would run to 64 lines:

$$[A \rightarrow (B \vee C)] \rightarrow [{\sim}D \,\&\, (E \vee {\sim}F)]$$

Is it a tautology? One way to find out is by reasoning backwards. Think again about the definition of tautology: it's a statement that comes out true on every line of a truth table. We could fill in all 64 lines of the truth table and see if they all come out true. But another way to proceed is to see *if it's even possible to make this statement come out false*. If you can find a way to make it come out false, then you've proven that it's *not* a tautology. How might you do that?

To start with, note that the main operator is an arrow.

$$\underset{\text{T}}{\rule{3cm}{0.4pt}} \rightarrow \underset{\text{F}}{\rule{3cm}{0.4pt}}$$

Is it possible for this whole statement to be false? Well, there's only one way that an arrow can be false: the chunk on the left would have to be true, while the chunk on the right is false. Now the chunk on the left looks like this:

$$A \rightarrow (B \vee C)$$

Is there some way to make this true? Sure! There are actually several ways. But consider the following assignment of truth values:

A: false
B: true
C: true

Any conditional with a false antecedent is automatically true. So because *A* is false the whole left-hand chunk comes out true—which is what we were looking for. Now what about the right-hand chunk? Can we make it false?

$${\sim}D \,\&\, (E \vee {\sim}F)$$

This is easy too. A conjunction is false whenever it has a false conjunct. So we can make the whole right-hand chunk false by making '$\sim D$' false. And we can do that by making *D* true.

D: true
E: true

F: true

It honestly doesn't matter what values we assign to E and F. As long as D is true, the right-hand chunk comes out false.

Now notice what we've done: the whole truth table for this statement would have 64 lines. But we've zeroed in on just one line:

A	B	C	D	E	F	$\sim D$	$\sim F$	$B \vee C$	$E \vee \sim F$	$A \to (B \vee C)$
F	T	T	T	T	T					

$\sim D \,\&\, (E \vee \sim F)$	$[A \to (B \vee C)] \to [\sim D \,\&\, (E \vee \sim F)]$

Feel free to fill in the rest of this line if you want. The process of reasoning here was just the reverse of what we did in §17 above. We started out by assuming that the target statement, in the rightmost column, is false, and then worked backward to see what truth value assignments to the letters would make that happen. Above, in §17, we were starting out with the truth value assignments to the letters and then working forwards to arrive at a truth value for the whole statement. Notice how this one line is all you need to prove that the statement in question is *not* a tautology. If it's false on even one of those 64 possible lines, that alone shows that it cannot be tautologous.

In this example, we just saved a huge amount of time in order to prove a negative. The statement in question is not a tautology. But what if it were? Can you also use the backwards truth table method to prove that something is indeed a tautology. You can indeed. To see how that might work, consider our old friend:

$P \to (Q \to P)$

Hopefully in §18 you filled in the full truth table to prove that this is a tautology. That table was no big deal—only four lines. But we can still use the backwards method here if we want. The main operator here is an arrow:

$$\underset{\text{T}}{\underline{\qquad}} \to \underset{\text{F}}{\underline{\qquad}}$$

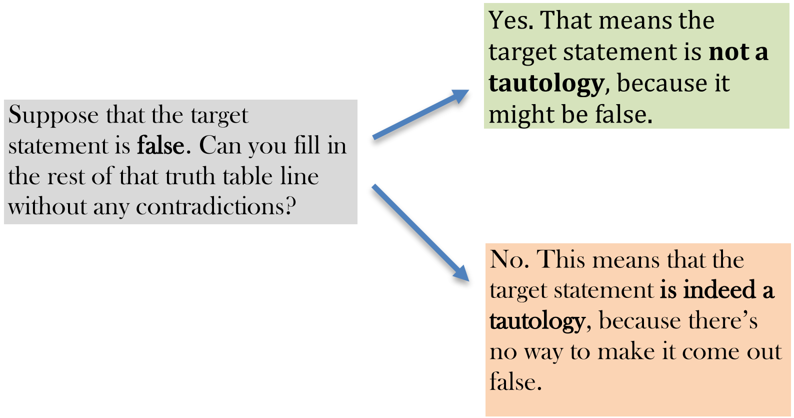
If this statement were not a tautology, then there'd have to be one line on the truth table where it's false. The only way to make a conditional statement come out false is for it to have a true antecedent and a false consequent. So far so good.

In this case, though, the antecedent is just P. So if the whole statement is false, we know that P must be true. Now let's look at the right-hand chunk:

$Q \to P$

We know that P must be true. But in order for the statement as a whole to come out false, the right-hand chunk must come out false as well. At this point, however, we run into a contradiction. If P were true, then ‘$Q \rightarrow P$ would also have to be true, according the truth table definition of the arrow. This clashes with what we've stipulated: if the whole statement is false—then ‘$Q \rightarrow P$ has to be false. So we have more or less hit the logical wall. We started out by saying: "Let's see if it's possible for the whole statement ‘$P \rightarrow (Q \rightarrow P)$ to come out false. If that is possible, then the statement is not a tautology." But things didn't work out! If we try to make the whole statement false, we inevitably run into a contradiction. Or here is another way to think about it: if the statement is a tautology, then there is no line on the truth table on which it comes out false. So if we start out by assuming that the target statement is false, things just won't work out, and we'll end up with a contradiction somewhere.

In case it helps, here is a flow chart rendering of the reasoning involved in the backwards method:



Suppose that the target statement is **false**. Can you fill in the rest of that truth table line without any contradictions?

Yes. That means the target statement is **not a tautology**, because it might be false.

No. This means that the target statement **is indeed a tautology**, because there's no way to make it come out false.

§23. The Anti-Laws of Logic

|Outline|

There are two types of statements in Langerese that are not tautologies. **Contingent statements** are statements that might be true, but might also be false. Or to put it another way, a contingent statement is true on at least one line of its truth table, and also false on at least one line. To give an easy example, an ordinary conjunction is contingent:

P	*Q*	*P*& *Q*
T	T	T
T	F	F
F	T	F
F	F	F

The conjunction is true on one line, and false on three. So it's contingent. Some statements, by contrast, are **self-contradictory**. This means that they come out false on every line of a truth table, much as a tautology comes out true on every line. For example:

P	~*P*	*P*& ~*P*
T	F	F
F	T	F

‘*P*& ~*P*’ is the classic example of a contradiction. It's what you might call a pure contradiction.

Much as tautologies are true in virtue of their form—or as we might say, formal truths—self-contradictory statements are false in virtue of their form. They are, as it were formal falsehoods. They are kind of like the anti-laws of logic. Interestingly, for every law of logic, there is a corresponding anti-law, and for every formal truth a corresponding formal falsehood. If you know that something is a tautology, all you have to do is put a ‘~’ sign in front of it to form a self-contradictory statement. And if you know that a statement is self-contradictory, you can place a ‘~’ in front of it to form a tautology, too. For example, ‘*P*& ~*P*’ is self-contradictory, but ‘~(*P*& ~*P*)’ is a tautology—the appropriately named **law of non-contradiction**.

In §18 and §22 we focused on truth tables and backwards truth tables for tautologies. By tradition, philosophers and logicians have been very interested in tautologies, going to great lengths to find them and prove them. It is even tempting to characterize a system of logic as a set of tautologies. But someone who is more interested in the dark arts of logic might focus instead on self-contradictory statements. (Why not?) Truth tables are also a perfectly good decision procedure for those. And you could use backward truth tables as well, but with one difference. Instead of starting with the supposition that the statement in question is false (which is what you do if you want to test whether it's a tautology), you'd have to flip things around and start with the supposition that the statement is true. If you could show, without contradiction, that the statement is true on at least one line of its truth table, then you would have shown that it is not self-contradictory.

Contingent statements are true or false depending on their *content*, not their form alone. If you want to know whether a contingent statement is true or false, then you have to figure out how to interpret it. That is, at a minimum, you have to make a decision about how to assign truth values to its component sentence letters. Thus, the laws and anti-laws of logic are true or false in virtue of their form alone, whereas contingent statements are true or false in virtue of *both* their form and their content.

One last interesting observation: Simple statements are always contingent. That's because a simple statement – say, P – can be either true or false. You can never have a truth table where P always comes out true, or one where P always comes out false.

Technical Terms

- Contingent statement
- Self-contradictory statement
- Law of noncontradiction

§24. Analyticity

|Outline|

The notion of a tautology is closely related to some other important philosophical concepts, and one of these is analyticity. In the late 1700s, the German philosopher Immanuel Kant drew a famous distinction between **analytic statements** and **synthetic statements**. The distinction was not entirely new; earlier thinkers, such as Hume and Leibniz, had anticipated it. But Kant codified it. In its original meaning, the term "synthesis" suggests that two things are getting put together. Kant held that a synthetic statement is a statement that combines two concepts together. For example:

The house next door is painted green.

This statement seems to combine two concepts: there is the concept of *the house next door*, and the concept of *being painted green*. The term "analysis," by contrast, suggests taking things apart. To analyze is to decompose or take apart. A classic example of an analytic judgment would be:

The cat next door is a mammal.

The concept *cat* and the concept *mammal* are not completely separate concepts, because being a mammal is already part of the definition of being a cat. Or as Kant sometimes put it, the concept *mammal* is already contained in the concept *cat*. So a judgment like the one above, rather than combining separate concepts, us merely taking apart—analyzing—the concept of a cat.

Another traditional way of thinking about analytic statements emphasizes definition. Many philosophers since Kant have held that analytic statements are true by definition. Anyone who understands the meanings of the words can appreciate that the statement is true. Synthetic statements are not like that at all. In order to know whether the house next door is painted green, you actually have to go look at the house to see what color it is. But now contrast the following two claims:

(1) The cat next door is a mammal.

(2) Either there is a cat next door, or there is no cat next door.

Claim (2) is just a familiar instance of the law of excluded middle—a tautology. But both claims seem true by definition, in a way. Claim (1) is true thanks to the definition of 'cat.' Claim (2), by contrast, is true thanks to the (truth table) definitions of disjunction and negation. So while both are true by definition, they are true by definition in different ways. Claim (2) is *formally* true by definition, whereas claim (1) is *materially* true by definition. Some philosophers will use the word ‘tautology’ for anything true by definition, which covers both claims (1) and (2). In logical contexts, however, it’s more common to reserve the term ‘tautology’ for claims such as (2)—see §18 above.

To help keep things clear, I propose saying that (1) and (2) are both analytic statements. This keeps with the tradition of saying that analytic statements are true by definition. But let’s call (1) a **materially analytic statement**, while (2) is a **formally analytic statement**. Tautologies in the logical sense (§18) are formally analytic statements.

Having said all that, it is also important to note that the analytic/synthetic distinction is also somewhat controversial. In a classic essay called “Two Dogmas of Empiricism,” the American philosopher and logician W.V.O. Quine argued that there is no good way to maintain a distinction between claims that are true by definition and claims that are true because of how the world is. Quine held that the difference between the claim that the cat next door is a mammal and the claim that the house next door is green is merely one of degree. It wouldn’t take much to get you to revise your belief that the house is green. But it would take a lot more—a major upheaval in our understanding of biology—to get you to revise your belief that the cat next door is a mammal.

Technical Terms

- Analytic statement
- Synthetic statement
- Formally analytic statement
- Materially analytic statement

§25.Testing for Logical Equivalence

|Outline|

Truth tables are a good decision procedure (though with some limitations, as noted in §20) that can reliably tell us whether a statement is tautologous, self-contradictory, or contingent. But truth tables are good decision procedures for other things, too. So far, we have focused on the properties of individual statements, but truth table methods are also useful for studying sets of statements. Truth tables can, for example, tell us whether two statements are logically equivalent. Back in §14, we looked at several ways of translating 'unless':

(1) p unless q

(2) $\sim p \rightarrow q$

(3) $\sim q \rightarrow p$

(4) $p \vee q$

It might not seem obvious at all, but (2), (3), and (4) are all logically equivalent statement forms. They are, at least when it comes to truth value, interchangeable. It's one thing just to assert that, but in logic, it's important to be able to prove these results that are not so intuitive. In this case, truth tables can help. (One other quick note here: I am using metalinguistic variables, or statement variables, instead of constants – see §2.)

To start with, let's construct two truth tables, for statements (2) and (3):

p	q	$\sim p$	$\sim p \rightarrow q$
T	T	F	T
T	F	F	T
F	T	T	T
F	F	T	F

p	q	$\sim q$	$\sim q \rightarrow p$
T	T	F	T
T	F	T	T
F	T	F	T
F	F	T	F

Take a moment to study these two tables. They are set up in the same way, with the same pattern of Ts and Fs under the 'p' and 'q' columns. And notice that the columns on the right (in green) have identical distributions of truth values. The two statements, (2) and (3) above, have the same truth value on each line of the truth table. If one is true, then so is the other. And if one is false, then so is the other. This means that they are logically equivalent. By the same token, if two statements ever have different truth values on one line of a truth table, they are not logically equivalent. For example 'p' is not logically equivalent to '$\sim p \rightarrow q$', because they have different truth values on the third line.

It might be really surprising to learn that statement (4) is equivalent to the others, but this is easy to show:

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

Recall that a disjunction is true whenever either of its disjuncts is true. So ‘$p \vee q$’ only comes out false on the last line—just like statements (2) and (3).

For the most part, we’ll use longer truth tables to test for logical equivalence. However, a shorter method is always available, which is a good thing, given how tedious it can be to fill in lots of T’s and F’s. In general, when using an abbreviated truth table test, you want to begin by supposing the opposite of what you want to test for. So for example, if you want to find out whether a statement is a tautology, start out by supposing it to be false, and then see if you can make that work without contradiction. Similarly, if you want to find out if two statements are equivalent, start out by supposing that they have different truth values, and see if you can make that work out without contradiction. Here’s how it would go in the present case. We have to look at two possibilities, because there are two ways in which the statements could have different truth values.

(4)	(2)
$p \vee q$	$\sim p \rightarrow q$
T	F
F	T

Does this work out? Start on the first line, on the right. If you have a false conditional statement, that means that ‘q’ has to be false, while ‘$\sim p$’ is true. If ‘$\sim p$’ is true, then ‘p’ is false. So in order for statement (2) on the right to come out false, both ‘p’ and ‘q’ have to be false. But if they were, then (4) would have to be false. There’s a contradiction there. So the first line doesn’t work out. Try the second line on your own.

Technical Terms

- Contradictory statements
- Formal contradiction
- Material Contradiction

§26. Quirks of Logical Equivalence

|Outline|

There is an interesting relationship between logical equivalence and material equivalence, the logical operation signified by the double arrow, '$\leftrightarrow$'. The easiest way to see this is to build on the examples from §25. There we saw that statements having the following forms are logically equivalent:

(1) $\sim p \rightarrow q$

(2) $p \vee q$

So what would happen if we treated (1) and (2) as chunks of a larger statement form, combining them with the double arrow?

(3) $(\sim p \rightarrow q) \leftrightarrow (p \vee q)$

Interestingly, statement (3) is a tautology! (I won't bother writing out the truth table for statement (3) here, but feel free to do so on your own in order to prove that (3) is tautologous.) It's fairly easy to see why (3) is tautologous. Look back at the truth tables—in §25—for statements (1) and (2). (1) and (2) have the same truth values on every line, which is just what logical equivalence means. In general though, when biconditional statements of the form '$p \leftrightarrow q$' come out true when 'p' and 'q' have the same truth value. That's just based on the truth table definition for the double arrow. So where 'p' and 'q' have the same truth value on every line, it follows that '$p \leftrightarrow q$' will be true on every line—a tautology! This reasoning works the other way, too. In general, whenever a statement having the form '$p \leftrightarrow q$' is a tautology, it follows that 'p' and 'q' will be logically equivalent. To capture this idea, let's say that statement (3) above is the **corresponding biconditional** for the logically equivalent statements (1) and (2). Indeed, every pair of logically equivalent statements has a corresponding biconditional.

As we develop our logical language, Langerese, we'll find that it has a number of fascinating quirks. Not every logic textbook makes much of a fuss about these, but I find that it's the quirks that really draw me back to logic over and over again. By quirks I just mean features of the system that seem really, really weird when you first notice them, but then make perfect sense once you reason them out. I'm also convinced that pausing to appreciate the quirks can really help one understand how logic works. One such quirk is the **logical equivalence of all tautologies.** Tautologies are so vastly different! There's everything from '$p \vee \sim p$' to '$(\sim p \rightarrow q) \leftrightarrow (p \vee q)$.' It seems weird to say that these two statement forms would be logically equivalent. And yet. If you work out the truth tables, you'll see that these two tautologies—along with every other tautology under the sun, always have the same truth value—namely, true! Always having the same truth value is the very definition of logical equivalence. All self-contradictory statements are logically equivalent to each other, too, for exactly the same reason.

The logical equivalence of all tautologies (and self-contradictory statements) does have one apparently strange consequence. Earlier (§14) we saw that there is a sense in which logically equivalent statements are interchangeable. Where two statements are equivalent, you can freely swap one for the other, without making any difference to truth values. To put this more precisely, we might talk about preservation of truth value: If you exchange one statement for its logical equivalent, you always preserve truth value. If what you started with was true, then what you'll end up with is true. If what you started with was false, then what you'll end up with is false. This matters for purposes of translation into Langerese: where you have two logically equivalent translations, it doesn't really matter (except for aesthetic or pragmatic reasons) which one you choose. But suppose someone says: "If Harry can cast spells, then Harry can cast spells!" –a tautology. You'd probably translate this as '$H \rightarrow H$'. But because all tautologies are logically equivalent, you could actually write down any tautology at all, and it would work. This fact is perhaps stranger than the idea of kids going to school to study spellcasting.

One thing that I dislike about some logic textbooks is that they introduce terms like 'material equivalence' without saying what the 'material' means, or what that terminology has to do with anything. In this context, 'material' means having to do with *content*. This is our old friend, the form/content distinction again. Consider the following pair of statements:

(4) I am older than my brother.

(5) My brother is younger than me.

Anyone can see that these are logically equivalent. As it happens, they are both true. But if the first one were false, then obviously the second one would have to be false as well. Notice, though, that neither of these statements contains any logical operators. Each is, as it were, a simple sentence. So if you wanted to translate them into Langerese, about all you could do would be to assign each its own letter. And because they are different statements, you'd have to use different letters—say, 'M' for statement (4) and 'B' for statement (5).

M: I am older than my brother.

B: My brother is younger than me.

These are logically equivalent, but you would never know that from doing a truth table:

M	B
T	T

T	F
F	T
F	F

This is the dumbest of all truth tables, but it's worth writing out just to make it really explicit that 'M' and 'B' do not show up as logically equivalent. They have different truth values on two lines. So what's going on? The issue here is very similar to what we encountered in the discussion of analyticity in §24. There we saw that there is a useful distinction between formally analytic truths (= tautologies) and materially analytic truths. Both are true by definition in some broader sense, but it's only the formally analytic truths that show up as tautologies of Langerese. In the same vein, statements (4) and (5) are **materially equivalent**, which is a way of saying that their logical equivalence is due to their semantic content. We can assert their equivalence using logic by writing '$M \leftrightarrow B$'. What makes them equivalent is the meanings of the terms "older" and "younger." Two statements are **formally equivalent** when their always having the same truth value is due to their logical form. So for example, '$p \vee q$' is formally equivalent to '$\sim p \rightarrow q$'.

Technical Terms

- Corresponding biconditional
- Formal equivalence
- Logical equivalence of all tautologies
- Material equivalence

§27. Contradiction

|Outline|

Two statements are **logically contradictory** when they never have the same truth values. If one is true, then the other is false. And if one is false, then the other is true. In the simplest case, ‘p’ and ‘$\sim p$’are logically contradictory. But sometimes, contradictions are more difficult to spot. For that reason, it can help to use truth tables. Consider:

$$P \rightarrow [\sim Q \vee (R \,\&\, \sim P)]$$

$$P \rightarrow Q$$

You can construct a truth table and then compare the lines under each statement to see if they ever have the same truth values. Note that in this case, one statement has three letters while the other has only two. This might seem to present an initial problem, as the truth table for the first statement would have 8 lines, while that for the second would have only 4. In order to get the comparison right, we actually have to do 8 line truth tables for both statements, as follows:

P	Q	R	$\sim Q$	$P \rightarrow Q$	$\sim P$	$R \,\&\, \sim P$	$\sim Q \vee (R \,\&\, \sim P)$	$P \rightarrow [\sim Q \vee (R \,\&\, \sim P)]$
T	T	T						
T	T	F						
T	F	T						
T	F	F						
F	T	T						
F	T	F						
F	F	T						
F	F	F						

This is the set-up. The highlighted columns are the ones we ultimately care about. The rest of the columns are just a matter of working our way up to the target columns, chunk by chunk. Here’s the complete table:

P	Q	R	$\sim Q$	$P \rightarrow Q$	$\sim P$	$R \,\&\, \sim P$	$\sim Q \vee (R \,\&\, \sim P)$	$P \rightarrow [\sim Q \vee (R \,\&\, \sim P)]$
T	T	T	F	T	F	F	F	F
T	T	F	F	T	F	F	F	F
T	F	T	T	F	F	F	T	T
T	F	F	T	F	F	F	T	T
F	T	T	F	T	T	T	T	T
F	T	F	F	T	T	F	F	T
F	F	T	T	T	T	T	T	T
F	F	F	T	T	T	F	T	T

These two statements, as it happens, are not contradictory. The truth table shows that it is possible for them to have the same truth value, because it's possible for them to both be true. Note that they both have the same antecedent, '***P***. A conditional statement with a false antecedent is always true. So if you stipulate that '***P*** is false, as on the last four lines of the truth table, both statements automatically come out true. So they're not contradictory.

Much of what we already said about logical equivalence also applies to logical contradiction. For example, there is a difference between material contradiction and formal contradiction. The following two statements are materially contradictory:

I am older than my brother.

My brother is older than me.

If the first one is true, then the second is false. And if the first one is false, then the second is true. (Even identical twins are not born at exactly the same time.) But in this example the contradiction is due to the meaning of "older," rather than to the logical structure of the two statements.

§28. Logical Consistency

|Outline|

So far, we've seen how to use truth tables on individual statements, to determine whether those statements are tautologous, contingent, or self-contradictory. Truth tables can also be used on *pairs* of statements, to determine whether they are logically equivalent or logically contradictory. There is, however, a more general question that we can ask about any set of statements whatsoever: Is it possible for all of them to be true? If so, then the statements are logically consistent. If it's not possible for all of them to be true, then they are inconsistent. These concepts are extremely important because philosophy is, by some lights, a quest to develop logically consistent views about things. That might seem to set the bar fairly low—surely logical consistency is easy to achieve. But it's actually one of the most difficult things in the world.

To start with, think about your belief system. Surely you want your beliefs to be true. At a minimum, then, that means that means you should try to avoid logical inconsistency. If there are any inconsistencies in your belief set, that means that at least one of your beliefs must be false. So the detection of inconsistency might be a prompt for revision. But it's extraordinarily difficult to test your beliefs for inconsistency. To make a go of it at all, you would have to know what your beliefs are, in the first place. As an exercise, try sitting down and writing a list of your beliefs. It (for me) go like this:

My dog's name is Toby.
Dinosaurs are extinct.
Greenland is cold.
I have many beliefs.
I just wrote some of my beliefs down.
Writing your beliefs down is really boring.
Etc. etc.

Let your mind wander a bit, survey some of what you think, and you'll appreciate that testing your whole belief system for logical consistency is a humanly impossible task, because you simply believe too much. If I could have just one superpower, I would wish to have a perfectly reliable, high-bandwidth ability to test for logical consistency. Imagine surveying your entire belief system at a glance, as well as any subset of your belief system, and being able to see immediately whether there is any logical inconsistency. Truth tables amount to a puny, human-sized version of this power, with all the limitations of human memory and attention. Still, truth tables are a wonderful tool to have at our disposal.

All the beliefs listed above are in fact true. (You can take my word for it about my dog's name.) Notice that they are also all simple statements. In general, simple statements are always formally consistent with each other. If you wanted to construct a truth table for the above list, you'd assign different letters to each statement. The table would technically have 2_{10} lines, since there are $n = 10$ statements in the list! It would have 1,024 lines. (Have fun filling that one in. The first column alone would have 512 T's and 512 F's.) But in any truth table, there's always a line—the top one, if you set it up conventionally, where every sentence letter gets assigned a T. Call this interesting feature of Langerese the formal consistency of all simple statements. But material consistency is a different matter. Consider:

> My only pet is a dog.
> My only pet is a guinea pig.

These statements are not contradictory. They can't both be true, but *they could both be false*. (Maybe my only pet is a parakeet.) If they could both be false, then they are not contradictory, because contradictory statements can never have the same truth value. Since they can't both be true, they are logically inconsistent. However, the inconsistency here (at least from the perspective of propositional logic) is material. These statements are simple ones, and the only way we could translate them is by giving each a sentence letter. This points to a limitation on the truth table technique: it can't detect material inconsistencies like this one, but only formal inconsistencies.

(A quick caveat: when we've gotten much further along in our development of Langerese, it will turn out that these statements do have logical structure, and that they can be translated using quantifiers—leap ahead to §48 if you wish.) However, the truth table technique doesn't work for quantified statements, and we're limited by the tools we have. If you want to use truth tables here, you'd get the result that the two statements above are formally consistent—which they are. They just aren't materially consistent.)

How might one use truth tables to test for formal logical consistency? Start with a set of statements:

$P \rightarrow {\sim}Q$

$R \leftrightarrow P$

${\sim}Q \rightarrow {\sim}R$

Then set up the truth table:

P	Q	R	${\sim}Q$	${\sim}R$	$P \rightarrow {\sim}Q$	$R \leftrightarrow P$	${\sim}Q \rightarrow {\sim}R$
T	T	T					
T	T	F					
T	F	T					
T	F	F					
F	T	T					
F	T	F					
F	F	T					
F	F	F					

The colored columns are the targets. The idea is to fill in the rest of the table, and then check to see if there are any lines where all three of the colored columns come out true.

P	Q	R	$\sim Q$	$\sim R$	$P \rightarrow \sim Q$	$R \leftrightarrow P$	$\sim Q \rightarrow \sim R$
T	T	T	F	F	F	T	T
T	T	F	F	T	F	F	T
T	F	T	T	F	T	T	F
T	F	F	T	T	T	F	T
F	T	T	F	F	T	F	T
F	T	F	F	T	T	T	T
F	F	T	T	F	T	F	F
F	F	F	T	T	T	T	T

As it happens, there are two lines on which all three of our target statements come out true: line 6 and line 8. In other words, there is at least one interpretation under which all three statements are true. Hence, they are logically consistent. We can make them all come out true by making ‘***P***’ false, ‘***Q***’ true, and ‘***R***’ false, for example (that’s line 6).

Logical consistency also has some interesting quirks that are worth pausing to think a bit about. For example, any set of statements that includes at least one self-contradictory statement is automatically logically inconsistent. That’s because self-contradictory statements can never be true. (Strangely, it is probably technically correct to say that a set containing a single self-contradictory statement is logically inconsistent.) Also, any set of statements that contains two logically contradictory statements is automatically inconsistent.

Something fun and perplexing to think about: Right at the beginning (§4), we made a decision to stick with tradition and adopt the principle of bivalence. In the meantime, we’ve introduced a slew of technical logical notions, giving them precise definitions, especially: tautology, self-contradictory statement, logical equivalence, logical contradiction, and logical consistency and inconsistency. We’ve defined all these notions based on the assumption that there are only two truth values, T and F. If we dropped bivalence and had more than two truth values—say, T, F, and X—would we need to redefine any of these concepts? If so, why? Would we need to introduce new concepts in addition to these? If so, what would those new concepts look like?

Technical Terms

- Formal consistency
- Formal consistency of all simple statements
- Logical consistency
- Logical inconsistency
- Material consistency

§29. How Many Logical Operators Do We Need?

|Outline|

This is an unusual question that many logic texts set to one side, or else leave for an appendix. But I take the view that it's really important to understand the "guts" of classical logic, and that means asking critical questions about it as we go along. At this point, we have some important tools in hand. The truth table technique, for example, gives us a way of proving logical equivalence. This alone raises some questions about the logical operators.

So far, we've gone along with tradition in adopting operators for negation, conjunction, disjunction, the conditional, and biconditional. If we really wanted to, however, we could scrape by with fewer than these. Suppose we wanted to strip things down as far as possible, and have the minimum number of logical operators.

We could pretty easily get rid of the double arrow. That's because:

$p \leftrightarrow q$ is logically equivalent to $(p \rightarrow q)$ & $(q \rightarrow p)$

We covered this in §16, but it should also be easy at this point to construct a truth table to prove that these are equivalent. Because they are equivalent, we can technically eliminate '↔' statements in favor of statements using only arrows and ampersands.

It's actually possible to eliminate arrows as well, because:

$p \rightarrow q$ is logically equivalent to $\sim p \vee q$

Again, you should have no trouble proving this with a truth table. What this means is that in principle, we could scrap both the '↔' and the '→'! Who needs them?

While we're at it, we could also get rid of '&', because:

$\sim p$ & $\sim q$ is logically equivalent to $\sim(p \vee q)$

So as a matter of principle, thanks to these logical equivalences, we could define away the '↔', '&', and the '→'. At the end of the day, if we so chose, we could get by with only two logical operators: negation and disjunction! As it happens, on a historical note, when Bertrand Russell and Alfred North Whitehead published their important work, *Principia Mathematica*, in the early 1900s, they used only disjunction and negation.

But on the other hand, logicians have proposed other operators, in addition to those introduced so far. One important example is the Sheffer stroke, which amazingly, enables you to strip things down even more, so that you don't even need negation. The Sheffer stroke basically means "not and." It is sometimes referred to as the "NAND" operator. So '$p|q$' means something like, "not both p and q."

p	q	$p \mid q$
T	T	F
T	F	T
F	T	T
F	F	T

But what about negation? Surely you can't have a viable system of logic without negation! Actually, the Sheffer stroke allows you to eliminate negation, because:

> $\sim p$ is logically equivalent to $p \mid p$

Feel free to test this on your own with a truth table, using the truth table definition of the Sheffer stroke above as a guide. With some additional cleverness, you can also eliminate wedges in favor of Sheffer strokes, because:

> $p \vee q$ is logically equivalent to $(p \mid p) \mid (q \mid q)$

This looks weird, but if you think it through, it makes sense. Remember that '$p \mid p$' is equivalent to '$\sim p$', while 'q|q' is equivalent to '$\sim q$'. So the expression on right is akin to saying '$\sim p \mid \sim q$'. But this is just that same thing as saying: "It's not the case that both '$\sim p$' and '$\sim q$' are true." And that is exactly what '$p \vee q$' says. We already saw just now that with negation and disjunction, you can get all the other operators via logical equivalence. If you can get negation and disjunction from the Sheffer stroke, then voila—the Sheffer stroke is technically the only operator you need!

Going forward, it's not essential that you learn the truth table for the Sheffer stroke. In fact, we're not going to use it at all. Why not? The important thing to see is that there are decisions (decisions that logic texts often make behind the scenes) which have to get made, and those decisions always involve trade-offs. One might think that the fewer the logical operators we have to work with, the better. After all, that means fewer truth table definitions to memorize. And there is also a gain in elegance and simplicity. So it might seem like there is a compelling case in favor of scrapping everything and just using the Sheffer stroke. Nevertheless, this minimalist approach has drawbacks, too. Simplicity always has a price. In this case, the price is that you end up with longer strings of symbols that are really hard to wrap your mind around. We might have fewer *kinds* of logical operators, but many statements (e.g. disjunctions, as above) will end up with larger numbers of operators. It takes only one wedge to write out a disjunction, but it takes three Sheffer strokes. That makes any sort of calculation or study more demanding. Another huge difference is that when we reason in English (or in any natural language), we actually use negation, disjunction, conjunction, and implication all the time. The nuances of "if ... then ..." in English are not perfectly captured by the truth functional arrow, but at least we can see that the arrow is in the neighborhood of what we usually mean by "if ... then ..." even while acknowledging various Procrustean challenges. By contrast, the Sheffer stroke doesn't capture our ordinary reasoning so well. So for example, you don't ordinarily hear someone say:

> "Next semester, I'm taking Dance NAND Philosophy."

You might say:

> "I'm not taking both dance and philosophy next semester."

But we have an easy and natural way to translate this without the Sheffer stroke:

> $\tilde{}(D \& P)$

In other words, there isn't any single operator in the natural language that corresponds with the Sheffer stroke. In ordinary life, we seem to use negation and conjunction.

As with all of these policy decisions about how to set up Langerese, the real question is what the purpose of Langerese is supposed to be. This is very much up for discussion. Pragmatically speaking, though, it seems plausible that many of the things we might want to do with our system of logic would be best supported by a system where the logical operators correspond somewhat well (though not perfectly) to operators in natural language.

§30. Validity

|Outline|

Right at the beginning, in §1, I said that logic is the study of arguments. And an argument, recall, is a set of statements, one or more of which (the premises) are alleged to provide support for the conclusion. It might seem strange to take so long before getting around to actually saying something about arguments. But my approach has been to take lots of time to lay the groundwork in a careful way. At this point, it will be really easy to shift from looking at individual statements and sets of statements to arguments, which are just a special kind of statement-set.

To start with, let's make it a policy to write arguments out in **premise-conclusion form**. This means numbering the premises, and using a line to mark off the conclusion:

(1) If hadrosaurs are dinosaurs, then hadrosaurs are extinct.
(2) Hadrosaurs are dinosaurs.
Therefore, (3) hadrosaurs are extinct.

Premise-conclusion form just makes the structure of an argument easier to see. This, recall, is the instance of *modus ponens* that we started with in §1. One reason for starting with *modus ponens* is that it is a valid argument form. In general, a **valid argument** is an argument where it is impossible for the conclusion to be false if all the premises are true. Validity is a hypothetical notion. In order to tell if an argument is valid, you have to ask: if the premises were true, would the conclusion have to be true as well? If so, then it's valid. If not, then the argument is invalid. To bring some earlier concepts to bear (§15), we might add that when an argument is valid, the truth of its premises is a sufficient condition for the truth of its conclusion. When we say that *modus ponens* is a valid argument form, all we mean is that every substitution instance of *modus ponens*—every argument with that form, is valid. Or to say the same thing in a slightly different way, when an argument is valid, it's not possible for all the premises to be true while the conclusion is false.

One thing that sometimes gives people trouble when they first study logic is that logic has a lot of technical vocabulary—terms like 'argument' and 'validity'—that overlaps with ordinary usage in confusing ways. It helps a lot to think of these terms as rather like scientific terms. For example, if you study cell biology, you'll learn terms such as 'ribosome' and 'chloroplast', all of which have precise, technical meanings. But those are also not terns that we use much in ordinary extra-scientific contexts. In logic, things are a bit different. The terms are just as technical, but we also use them in nontechnical ways all the time; they have established meanings already. For example, if you say that somebody "makes a valid point," or "has a valid argument," in ordinary contexts, that might just be a way of expressing your own approval or agreement. But you're not necessarily saying anything about the formal structure of the argument. To do logic well, you have to train yourself to set aside these ordinary meanings. For example, it has been years since I've told anyone, "you have a valid point," because I try to be very precise when I use the term 'validity.' A related problem arises when we consider that many arguments that are valid in the technical logical sense are not ones that any sane person would every want to approve or agree with. We'll look at some examples of valid arguments—in the technical, logical sense—that no one in their right mind would regard as making a "valid point" in the ordinary sense.

An argument is invalid when it's possible for the conclusion to be false even if all the premises are true. Here's an example of an invalid argument.

(1) If Harry Potter is a Slytheryn, then he owns a wand.
(2) Harry owns a wand.
So therefore (3) Harry Potter is a Slytheryn.

The conclusion is false. (In case the reference is unfamiliar, Harry actually belongs to House Gryffyndor, rather than House Slytheryn.) The two premises of this argument, seem true. Harry does own a wand. It's also true that if he's a Slytheryn, he would own a wand, since all students at Hogwarts presumably have wands. But the conclusion is false.

T (1) If Harry Potter is a Slytheryn, then he owns a wand.
T (2) Harry owns a wand.
F So therefore (3) Harry Potter is a Slytheryn.

If it's possible at all for an argument like this to have true premises and a false conclusion—as this one actually does!—then it's invalid, because actuality is a proof of possibility. In fact, the argument above is a notoriously sneaky one, an invalid argument form known as affirming the consequent. It's sneaky because it looks a lot like *modus ponens*, but with one crucial difference: in *modus ponens*, you affirm the antecedent, not the consequent. Because affirming the consequent is an invalid argument form, that means that every instance of the pattern—every argument with this form—will be invalid.

It's extremely important to be aware that an invalid argument doesn't actually have to have a false conclusion! This is because validity is a hypothetical notion. The relevant question is always: is it *possible* for the conclusion to be false, even while all the premises are true? There are lots of invalid arguments where every statement in the argument is true. Here's another instance of affirming the consequent:

T (1) If Toby has to go pee, then he is ready for his walk.
T (2) Toby is ready for his walk.
T So therefore, (3) Toby has to go pee.

Apologies for the dumb example involving my dog! Hopefully it's easy to see how all three of these things might be true. In fact, I think they are true as I write this. Toby seems ready for his walk, and I'm sure he has to go pee. The conditional premise (1) is definitely true. BUT this argument is still invalid. It is an instance of affirming the consequent (note that the consequent of the conditional premise is highlighted). The important thing, though, is that if we think it through carefully, we can imagine a possible situation where (1) and (2) are true, but the conclusion is false. Suppose Toby just peed an hour ago, so his bladder is fine. But he's bored and he really wants to go play. For that reason, perhaps, (2) is true: he's ready for his walk. And (1) is still true as well. But in this alternative scenario, the conclusion (3) would be false—he doesn't really have to pee. Instead he's just bored and antsy.

What this example shows is that validity is not really about having a true conclusion. Instead, validity is about the *relationship* between the premises and the conclusion of an argument. Do the premises logically imply the conclusion? In the Toby example, the answer is no. The conclusion might be true—indeed, it is—but it does not follow logically from the premises.

It's also possible to have valid arguments with false conclusions. This is a really counter-intuitive point, so it is worth pausing to think through. Many people naturally assume that if an argument is valid, its conclusion must be true. This is completely wrong. A valid argument can have a false conclusion if one or the other of the premises is also false. Here's an example of a *modus ponens* argument with a false conclusion:

> (1) If Socrates was convicted of corrupting the youth of Athens, then he had a pet dinosaur.
> (2) Socrates was convicted of corrupting the youth of Athens.
> So therefore, (3) Socrates had a pet dinosaur.

Obviously the conclusion is false. But this *modus ponens* argument is still valid, because IF the premises (1) and (2) were both true, then the conclusion (3) would have to be true as well. Again, what matters is not whether the premises or the conclusion are in fact true, but rather the relationship between them.

Truth tables are a good decision procedure for validity. Here is a truth table for *modus ponens*.

Premise	Conclusion	Premise
p	q	$p \rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

This should be familiar. In fact, it's the same as the truth table definition for the conditional statement. (When you think about it, that makes sense, since the only logical operator in a *modus ponens* argument is the arrow.) The difference, though, is in how you look at the truth table. If we're just defining the arrow operator, then obviously the column we should care most about is the one on the right. When checking an argument for validity, though, you have to look at multiple columns—in this case, all three. The question is: is there any line on which the conclusion is false while all the premises are true? If so, then the argument is invalid. If not, then it's valid. Again, the question we're asking when it comes to validity is whether it's possible to have all true premises and a false conclusion. If yes, then the argument is invalid. Here, though, the answer is no. The conclusion is F on line 2 and line 4. But on each of those lines there is also a false premise. So the argument is valid.

Things look different, though, with affirming the consequent:

Conclusion	Premise	Premise
p	q	$p \rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

Notice that this is exactly the same truth table! It might seem very strange that a valid argument and an invalid argument can have the same truth table. The reason for that is that both *modus ponens* and affirming the consequent are exactly the same set of statements. If we merely defined an argument as a set of statements, there would be no difference between them—an interesting fact. Consider:

{'p', 'q', '$p \rightarrow q$'}

But an argument is more than just a set of statements. It's a set of statements, where some of those statements are alleged to stand in a certain sort of relationship to another member of the set. The difference between *modus ponens* and affirming the consequent has everything to do with that alleged relationship., and with the question of which statement is alleged to follow logically from which others. The conclusion of an affirming the consequent argument is 'p'. So in the above truth table, we have to ask whether it's possible for the conclusion, 'p' to be false while both the premises are true. And here the answer is Yes—on line 3. So the argument is invalid. This same technique can be used to test the validity of any argument in Langerese, though subject to the human limitations discussed in §21.

One potentially disorienting thing about logic is that we naturally tend to suppose that "valid" means "good." Isn't a valid argument a good argument? As it happens, though, validity is neither necessary nor sufficient for having a good argument! This is one of the stranger things about logic, but it's an extremely important point. Some valid arguments are bad arguments, while some invalid arguments are good! I'll return later to the issue of how an invalid argument can be good. And we've actually already had an example of how a valid argument might go badly, with the argument above about Socrates's pet dinosaur. Here's another terrible modus ponens argument, just for good measure.

(1) If Gryffyndor won the quidditch match last week, then Bigfoot studied logic.
(2) Gryffyndor won the quidditch match last week.
(3) Therefore, Bigfoot studied logic.

This argument is valid. But if I tried to use it to convince you that Bigfoot is a logician, you should, um, laugh. Premise (1) is just silly. Hopefully this dumb example will help you to remember, always, that a valid argument is not the same thing as a good argument.

Consider now another example of a valid argument:

(1) I am 45 years old.
(2) My brother is 42 years old.
(3) Therefore, I am older than my brother.

This argument is also valid. If premises (1) and (2) are true, then the conclusion (3) must also be true. However, if you tried to translate the argument into Langerese, it would look like this:

(1) M
(2) B
(3), therefore O.

This isn't valid at all. If we constructed the truth table for this argument, there would be one line where 'M' and 'B' are true, while 'O' is false. So it seems like something has gone wrong—like the validity of the argument got lost in translation.

The solution to this puzzle is to draw a distinction between material validity and formal validity. This parallels our earlier distinction between materially analytic statements and formally analytic statements, or tautologies (§24). Sometimes, validity depends entirely on the logical form of an argument. In those cases, we can use truth tables as a reliable decision procedure. Sometimes, though, the validity of an argument depends on what the statements actually say, or on the content of the argument. Truth tables give us a way of testing only for formal validity, not material validity. So if you have an argument that's materially valid, how can you translate it into Langerese in such a way that its validity will show up? The way to do this is to add a premise:

(1) M
(2) B
(3) $(M \& B) \rightarrow O$
(4) Therefore, O

This argument is actually formally valid, and its validity is easy to show on a truth table. All we've done is to add premise (3). Premise (3), '$(M \& B) \rightarrow O$' is what is known as the corresponding conditional of the original argument. To form the corresponding conditional, you just make the conclusion of the original argument the consequent of the conditional, while conjoining the premises to form the antecedent.

(premise & premise) → conclusion

Every argument has a corresponding conditional statement. For example, the corresponding conditional of the new argument we just formed would be:

$$\{M \,\&\, [B \,\&\, [(M \,\&\, B) \rightarrow O]]\} \rightarrow O$$

All of this points to an interesting philosophical question, which the writer Lewis Carroll (author of *Alice in Wonderland*) explored in a famous essay that takes the form of a dialogue between Achilles and the tortoise. Carroll wondered: If there was a need to add the corresponding conditional statement as a new premise to the original, materially valid argument, then do we *always* need to add the corresponding conditional? If so, that would seem to lead to an infinite regress. For when you add the corresponding conditional as a new premise, you've created a new argument with a different (& longer) corresponding conditional of its own. One response to Lewis Carroll's question is to say that we're interested in formal validity. If you take a formally valid argument, add its corresponding conditional as a premise, the new argument will also be formally valid. But since you had a formally valid argument to begin with, that additional move isn't really needed.

Many logicians would say that formal validity is really what logic (or at least, formal logic) is all about. Some would say that the central issue in logic is the concept of logical consequence. What does it mean for one statement to follow logically—or to be a logical consequence—of others? "Validity" is really just another name for logical consequence. For we can say that an argument is formally valid if and only if its conclusion is a logical consequence of its premises.

Technical Terms

- Validity
- Invalidity
- Formal validity
- Material validity
- Corresponding conditional
- Affirming the consequent
- Premise-conclusion form

§31. Validity's Quirks

|Outline|

Validity is one of the most important concepts in logic. But it is also quite strange, and it's important to be open and up front about the strangeness. Recall from §9 that material implication is strange and counterintuitive, too. Now we just saw in §30 that you can take a materially valid argument and make it formally valid by adding in the corresponding conditional as a new premise. This suggests that there is a tight relationship between material implication and validity. But that also means that a lot of the counter-intuitiveness of material implication will bleed over into discussions of validity. So we should take some time to think through validity a bit more.

One interesting feature of validity to start with: The corresponding conditional of any valid argument is a tautology. In order to see why this is so, consider the corresponding conditional for modus ponens:

$$[p \,\&\, (p \rightarrow q)] \rightarrow q$$

This comes out true on every line of its truth table:

p	q	$p \rightarrow q$	$p \,\&\, (p \rightarrow q)$	$[p \,\&\, (p \rightarrow q)] \rightarrow q$
T	T	T	T	T
T	F	F	F	T
F	T	T	F	T
F	F	T	F	T

By the same token, any time you have a conditional statement that's a tautology, it corresponds to a valid argument form. Testing to see if a conditional statement is a tautology and testing to see whether an argument is valid are this, in a way, two sides of the same coin.

Some valid arguments are extraordinarily weird. Consider the following:

(1) Socrates was Athenian.
So therefore, (2) either Socrates was Athenian or there is a dinosaur in my office.

The strange thing about this is that Socrates being Athenian has nothing whatsoever to do with whether there is a dinosaur in my office. But try symbolizing the argument:

(1) S
(2) $S \vee D$

And then do the truth table:

S	D	$S \vee D$
T	T	T
T	F	T
F	T	T
F	F	F

The column for the premise is in blue, while that for the conclusion is shaded in green. Is it possible for the conclusion to be false while the premise is true? No. The conclusion is only false on line 4, but the premise is false on that line too. If the premise is true, the conclusion must be true as well. This argument just exploits a general feature of disjunction: Having one true disjunct guarantees that the disjunctive statement is true.

Here's another example of an argument that's weird, but valid:

(1) q
(2) Therefore, $p \rightarrow q$

Thus:

(1) Socrates was Athenian.
(2) So, if there is a dinosaur in my office, then Socrates was Athenian.

This, again, seems ridiculous. But a simple truth table proves that this argument is valid. The trick is to see that a conditional statement with a true consequent is always true, no matter what the antecedent says, and even if the antecedent is totally irrelevant to the consequent. It actually gets even worse than this when you consider arguments such as the following:

(1) Socrates was Athenian.
(2*) So, if Socrates was Spartan, then Socrates was Athenian.

This argument is still formally valid, though there is something really weird about it. The weird thing has to do with the *content* of the antecedent and the consequent of (2*). You might be thinking: If Socrates was Spartan, how could he be Athenian? That makes no sense, since Athens and Sparta were different cities. This is a case where the antecedent of the conclusion seems materially inconsistent with the consequent. Given what the antecedent and the consequent say, it's hard to see how they can both be true. Just remember that formal validity has nothing to do with what the statements in question actually say. The only thing that matters for formal validity is the logical relationship between the premise(s) and the conclusion.

Another strange feature of validity is that every argument with a tautology for a conclusion is valid. Consider the following example:

(1) Titanosaurs were sauropods.
(2) Sauropods were herbivores.
(3) Therefore, either New London is in Massachusetts, or New London is not in Massachusetts.

This argument is a head-scratcher. At first, it looks like the premises might be headed to a different conclusion, perhaps that Titanosaurs were herbivores. The actual conclusion (3) has nothing whatsoever to do with either of the premises. But the argument is nevertheless valid! That's because (3) is a tautology. If you formalize the above argument, it looks like this:

(1) T
(2) S
Therefore, (3) $M \vee {\sim}M$

You might think, "Wait, how can this be valid, since the letter '*M*' doesn't even appear in the premises anywhere?" The trick is to see that the conclusion cannot possibly be false. That means, in turn, that it's not possible for the premises all to be true while the conclusion is false. But that's the very definition of 'validity.' If it helps, here's the truth table:

T	S	M	${\sim}M$	$M \vee {\sim}M$
T	T	T	F	T
T	T	F	T	T
T	F	T	F	T
T	F	F	T	T
F	T	T	F	T
F	T	F	T	T
F	F	T	F	T
F	F	F	T	T

Now remember the definition of 'validity': if there is one line where the premises (blue and brown) are all true while the conclusion (in green) is false, then the argument is invalid. Otherwise, its valid. Strange as it seems, this one is clearly valid.

One feature of validity that these reflections really drive home is that validity has nothing to do with relevance. Relevance is a matter of content: it means that what the premises say has something to do with what the conclusion says. The above example shows that an argument can be valid even when the premises are totally irrelevant to the conclusion. This is one respect in which validity fails to track ordinary intuitions about what makes something a good argument.

So an argument whose conclusion is a tautology is always valid. At the same time, an argument with inconsistent premises is always valid. This, too, has to do with the very definition of 'validity.' If the premises are inconsistent, that means it's not possible for them all to be true (§28). But if it's not possible for the premises all to be true, then it’s not possible to have all true premises and a false conclusion—so the argument is valid by definition. Here’s an example:

(1) P
(2) $\sim P$
Therefore (3) Q

Note the failure of relevance again. One interpretation of this argument might be:

(1) Harry Potter is a wizard.
(2) Harry Potter is not a wizard.
Therefore (3) There is a dinosaur in my office right now.

Because the premises are inconsistent, it doesn’t really matter what the conclusion says. Here’s how the truth table looks:

P	$\sim P$	Q
T	F	T
T	F	F
F	T	T
F	T	F

Because there’s no line on which ‘P and ‘$\sim P$ both come out true, there’s no line where you have all true premises and a false conclusion. So the argument is valid. The general rule is that an argument with inconsistent premises is always valid. If an argument has a premise that’s self-contradictory, then it also follows that the argument is valid, because a single self-contradictory premise makes the premise(s) logically inconsistent.

Incidentally, this last phenomenon is known as logical explosion. The idea is that an inconsistent set of statements will (sort of, in a way) logically explode, because anything and everything follows logically from the inconsistent set. If you start with an inconsistent set of statements, you can use them as premises in a valid argument for anything! This might seem like a minor puzzle, but it can actually cause some headaches. Consider, for example, the following argument:

P1. Everyone’s belief system includes some (often minor) logical inconsistencies.

P2. In general, you ought to believe the logical consequences of your beliefs. For example, if you believe ‘P as well as ‘$P \rightarrow Q$,’ then you ought to believe ‘Q’.

C. Therefore, everyone ought to believe everything.

The conclusion of this argument is obviously crazy. It implies that everyone should believe that the moon is made of green cheese, that Mars is made of Styrofoam, etc. So it's an interesting philosophical question just how to avoid the conclusion. Unfortunately, just speaking for myself, I am not terribly confident that my belief system contains no inconsistencies. So I might think about avoiding the conclusion by modifying or revising P2. Anyhow, I will leave it to you to think for yourself about how best to handle this argument.

- Explosion

Technical Term

§32. Deductive Arguments

|Outline|

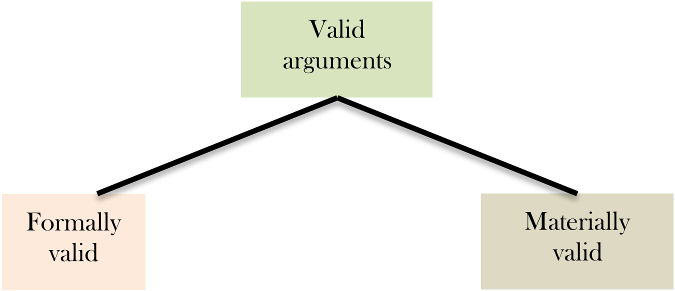
Earlier I noted that validity is neither necessary nor sufficient for having a good argument. Some valid arguments are just not good ones. For example, they might have false or unjustified premises. And we'll see later on that some invalid arguments are perfectly good ones. In light of this, you might be forgiven for wondering what all the fuss is about over validity. Why would some logicians even define formal logic as the study of validity, given the somewhat tenuous connection between the notion of validity and the notion of a good argument?

But even if validity is neither necessary nor sufficient for having a good argument, we often *try* to make arguments that are valid. Sometimes we succeed; other times we don't. But if people are trying to make a valid argument, or if we have an interest in arguments being valid, then all of a sudden validity begins to look really important. When I say that people often try to make valid arguments, I have to be really careful. Those who haven't studied logic may have no idea what validity even means. And how can you try to make a valid argument if you don't know what validity is? Nevertheless, we often reason in the following spirit: "This thing is true, and this other thing is true. So, you just have to agree that *P*." When someone argues in this spirit, you might say that they are trying to make a valid argument, even if they have no idea at all what "validity" means. If you want to give justification for some claim, or to give reasons for believing it, then one way (though not the only way) to do that is to try to give a valid argument for the claim in question. For this reason, validity really is worth studying.

In general, when someone is *trying* to make a valid argument, or (even if they don't know what validity is) trying to show that we should accept a claim because it is a logical consequence of some other claims, let's call that argument **deductive**. And let's say that a person is engaging in deductive reasoning when they are trying to make valid inferences. Of course, this means that some deductive arguments—the successful ones—*are* valid, but many are not. The whole point of talking about trying is to open up the possibility of failure. This way of defining deduction has one interesting consequence: it turns out that whether an argument is deductive or something else (such as inductive—a notion that we'll look at shortly) depends entirely on wat the arguer is trying to do. Sometimes this is really obvious from context, but in some cases, it can be very difficult to tell. In a case where an argument is indeed valid, we can generally say, pretty safely, that it's deductive. Even if the person wasn't trying to make a valid argument, they may as well have been. And it might be charitable to interpret them that way (§37).

Things get more complicated when someone makes an argument that's invalid. Invalid arguments are, by definition, bad deductive reasoning. But that just falls out of the above definition of deduction. When someone makes an invalid argument, we often have to make challenging interpretive decisions. Maybe they were trying to make a valid argument and failing—bad deductive reasoning. Or maybe they were trying to make some other kind of argument. Or maybe they were not really trying to make an argument at all. We'll return to discuss these interpretive challenges a bit more later on, but one thing to consider is that it is often helpful to be able to tell *first* whether an argument is valid. Knowing this first can provide some clues as to how to interpret what's going on.

I want to highlight one strange thing about the way I have defined 'deduction'. Much like botany or zoology, logic is very much about classifying arguments. In fact, we've done some classification already:



We've also looked at some subcategories of formally valid arguments, like *modus ponens*, and arguments with inconsistent premises. The different types of formally valid arguments are all recognizable by their logical form. However, in order to tell whether an argument is deductive, you really have to look beyond the logical structure of it, and even beyond the material content of the statements that comprise the argument; you have to look into the head of the arguer, to see whether that person is *trying* to make a valid argument. And depending on the context, this can be difficult to discern. Think about all the times in life when you've tried to do X, but failed, and then revised your plans downward, thinking, "Oh, I never really cared about X, I was really trying to do something else." For example: your team loses the soccer match, and you say, "Oh, it would have been nice to win, but we really just wanted to go out there and enjoy the game." The point is just that figuring out whether an argument fits the pattern of *modus ponens* is really easy, because the logical form is easy to see. But figuring out whether an argument is deductive can sometimes be a messy, contextual affair.

Deductive arguments are usually contrasted with **inductive** ones. So far, we've defined a deductive argument as one in which the arguer is asserting (perhaps only implicitly) that if the premises are true, then the conclusion must also be true—or one where the arguer is *aiming for validity*. We can use the term "inductive" to refer to *any non-deductive argument*. That is, anytime someone uses a set of statements to justify or support some other statement/conclusion, without suggesting that the conclusion is a logical consequence of the premises—or without aiming for validity—the argument is inductive. Thus, the term 'inductive' is a kind of catch-all for arguments that aren't deductive.

It might be worth pausing just to note the definitional strategy that I am using here. I started out with validity, which is easy to define (§30). Then I defined ‘deductive argument’ in terms of validity. Then, last of all, we can define ‘inductive argument’ as any argument that’s not deductive.

There are lots of examples of inductive arguments from ordinary life. Here is a simple one:

(1) In recent weeks, Shiloh has usually gotten to the park around 8:30 am.
(2) Therefore, if we go to the park tomorrow around 8:30, Shiloh will be there.

The first thing to notice about this argument is that it’s invalid. Even if the premise (1) is true, there could be other reasons why the conclusion (2) is false. Perhaps Shiloh’s humans will be sick tomorrow. Or maybe they are going on a trip. Lots of different factors could keep Shiloh away from the park tomorrow. Nevertheless, the premise (1) obviously does lend some support to the conclusion. It would seem that if (1) is true, then (2) is *probably* true as well. The fact that this argument is invalid does not mean that it’s a bad one. On the contrary, if you make an argument like this one, then you are probably not even trying to make a valid argument. Instead, you are just considering weaker reasons that might support a conclusion.

To make this a little clearer, let me give an example from philosophy of science.

(1) If Einstein’s general theory of relativity is correct, then scientists should be able, under just the right conditions, to observe the bending of a ray of light coming from a distant star as it passes by a massive body.

(2) In 1919, a team led by Arthur Eddington actually did observe the bending of a light ray during a solar eclipse.

(3) Therefore, Einstein’s theory of general relativity is true.

This type of argument is actually pretty common in science. But alas, it’s invalid! If you formalized it, it would look like this. I’ll use ‘T’ for “Einstein’s theory is correct,” and ‘O’ for “you observe light bending under the right conditions.”

(1) $T \to O$
(2) O
(3) T

This, sad to say, is our old friend affirming the consequent, which is demonstrably invalid! So if a scientist were to make an argument like this, what should we say? One option is to say that it’s a deductive argument, but a bad one. Another option is to say that the scientist isn’t even really trying to make a deductive argument here. We can translate it into Langerese if we want to, and even test it for validity, but making a valid argument is not really the goal. Most philosophers of science go with the second of these interpretations, in part because the first one makes science seem irrational.

One more thing: There is an old-fashioned way of thinking about deductive arguments that is (if you ask me, anyhow) misleading and unhelpful. If you read older works in philosophy, you might come across the idea that deduction involves reasoning from general to particular. Inductive reasoning, by contrast, goes from particular to general. People who take this view are usually thinking of examples like the following:

(1) All wizards have wands.
(2) Hermione is a wizard.
(3) Hermione has a wand.

Premise (1) is a general statement about all wizards, while the conclusion (3) is a particular statement about just one person. Now contrast that with the following:

(1) Hermione has a wand.
(2) Harry has a wand.
(3) Ron has a wand.
(4) Dumbledore has a wand.
(5) All wizards have wands.

Notice that all the premises of this second argument are particular statements about particular people, while the conclusion is a general statement about all wizards. The old-fashioned view would treat this as an inductive argument because it goes from particular to general.

Deduction = Reasoning from general to particular

Induction = Reasoning from particular to general

Wrong

I mention the old-fashioned view here only because I want to reject it and explain why it's wrong. One problem is that there are counterexamples—examples of valid argument forms that do *not* go from particular to general. One easy example, is an argument form known as disjunctive syllogism. It's called that because the premise doing the work is a disjunction. Also, for the record, a syllogism, is an argument with exactly two premises.

(1) Either Harry will take potions next semester, or he will take defense against the dark arts.
(2) Harry will not take potions next semester.
(3) Therefore, Harry will take defense against the dark arts.

Which has the following form:

(1) $P \vee D$
(2) $\tilde{} P$
(3) Therefore D

Feel free to use a truth table to verify the validity here, or you can just take my word for it. The point is that when people make valid arguments like this, they are plausibly reasoning deductively. Indeed, most logicians regard this as a classic example of a valid deductive argument form. However, the reasoning here does not go from particular to general, because the argument has no general statements at all. There are other reasons to reject the old-fashioned view, but this example alone is utterly decisive. Here is a better way to think about the difference between induction and deduction:

Deduction: What you're doing when you *try* to make a valid argument, that is, when you try to justify a conclusion by showing that it if a certain set of premises is true, then the conclusion must also be true.

Induction: What you're doing when you try to justify a conclusion by showing how some premises support it, but without claiming that if the premises are true, then the conclusion must be true.

Another way of stating the main take-home message of this section is that although we do have a wonderfully reliable decision procedure for validity (namely, truth tables), there isn't any similarly reliable decision procedure for telling whether an argument is deductive or not. That requires more of a judgment call.

One last thing is worth noting. Nearly all of the logic covered in this text is deductive. Except for a brief discussion coming up in §35, I will have very little to say about induction. Our focus in this book is on logical form. Although inductive arguments are well worth studying, and although they come with a variety of fascinating structural patterns all their own, they are, let's just say, a lot messier.

Technical Terms

- Deductive argument
- Inductive argument

§33. Soundness

|Outline|

Validity is entirely a matter of the *relationship* between the premises of an argument and its conclusion. This is why you have to think about validity hypothetically: If the premises were all true, would that guarantee that the conclusion is true as well? But we've already looked at some examples where valid arguments are bad arguments in the end. In those cases, the problem is not about the relationship between the premises and the conclusion at all; the problem, rather, is that the premises are false, or (in some of the above examples) stupid, or just not well supported by evidence. Validity, as noted earlier, is not a sufficient condition for having a good argument. For a good argument, more is needed.

Logicians traditionally define a **sound argument** as a valid argument with all true premises. On this definition, soundness includes validity, but also requires something more. This traditional definition of soundness means that a sound argument is basically airtight: if the premises are all true, and if the conclusion logically follows from the premises, then the conclusion has to be true as well. Because soundness includes validity, a sound argument will automatically have a true conclusion. Soundness is, in a way, the gold standard of argumentation. If you have a sound argument, there isn't really any way to make your argument better. But it's also important to keep in mind that lots of good arguments fall short of soundness. Some invalid arguments, we'll see, are still good arguments; and invalidity automatically renders an argument unsound.

In some cases, it's pretty easy to tell when you have a sound argument. Here's one:

(1) If Toby is a dog, then Toby is a carnivore.
(2) Toby is a dog.
(3) Therefore, Toby is a carnivore.

The premises are both true (Toby is, in fact, my dog). And the argument is valid, having the form of *modus ponens*. So it's sound. But you might notice that it's also really boring. In fact, most sound arguments will turn out to be pretty boring, because the premises will have to be either common knowledge, like premise (1), or else really easily verifiable, like premise (2). Once we clarify who we mean by 'Toby', no sensible person would disagree with either (1) or (2). Many logic textbooks make a big deal of soundness, as if it were what we're really aiming for in logic. But when we shift to more interesting arguments—arguments where the premises are more contested, soundness turns out to be very, very difficult to achieve. Validity is much easier.

In order to make this clear, suppose I gave you an instruction: In all the rest of the courses you take at Connecticut College (or wherever you are currently studying), you must never, ever, ever, write a paper that contains an invalid deductive argument. Call this the **formal validity requirement**. Notice how carefully I worded that: You're allowed to make formally invalid arguments sometimes, but you're never allowed to make invalid *deductive* arguments. (It's okay to make invalid inductive arguments.) Given earlier definitions, that is a way of saying: If you try to make a valid argument and fail, there's no excuse! The reason why there's no excuse is simple: we have an excellent decision procedure for validity in the form of truth tables. When in doubt, you can easily translate your argument into Langerese and use a truth table to make sure it's valid. (Things are complicated just a bit by the fact that truth tables only test for formal and not material validity; that's why it's only a formal validity requirement.) Anyhow, once you get good at logic—and you will—meeting the formal validity requirement turns out to be really easy. But suppose I issued a soundness requirement: You must never, ever, write a paper containing an unsound deductive argument. That would be a totally unreasonable demand, and if I made such a demand, you would have every right to be annoyed. For in order to live up to the soundness requirement, you would have to make sure that you never, ever, make an argument with a false premise. And how are you supposed to do that? Importantly, we have a decision procedure for validity in propositional logic—that is, in Langerese as we've developed it so far—but there's no decision procedure for truth. In order to assess the truth of your premises, you need to look at what they say—i.e., their *content*.

In many cases, it's just hard to tell whether premises are true. We might lack the relevant background knowledge. Or wonder whether our sources are reliable. Other reasonable people might disagree with us, in which case we'd have to examine further evidence and arguments. Insisting that an argument have all true premises sets the bar extraordinarily high. For what it's worth, my own take on things is that soundness is usually defined in a way that sets the bar *too* high. In cases of interesting, substantive disagreement in science, or ethics, or whatever the context might be, it will often be really hard to tell whether our arguments are sound. Soundness is nice when you can get it, but it seems a little crazy to demand it all of the time. But we can lower the bar in the following way: suppose we define a **semisound argument** as a valid argument whose premises are all *justified*, or well supported by evidence and arguments. (For the record, "semisound" is a term I just made up; it's not a recognized technical term in logic. Just what it takes to justify a premise is a huge question—the subject of much work in epistemology, or the theory of knowledge). So let's say that a good deductive argument is a semisound argument: a valid argument, where you are able to provide adequate justification for each premise, some reason to believe each premise. Of course the justification you offer for each premise ought to have some connection to truth, in the sense that it ought to provide some reason for thinking that the premise is true. So soundness might still be lurking in the background as a cognitive goal. But semisoundness will be easier to achieve. Some semisound arguments might be unsound. Sometimes, a premise might turn out false even though we have what we think is a pretty decent justification for it. Some sound arguments will also be un-semisound (with apologies for the atrocious terminology). There might for example be cases where an argument has true premises, but we have no evidence for them and no idea how to justify them.

Testing an argument for validity is relatively easy; as long as we're focusing on validity, we are still in the pristine realm of formal logic. But once you start assessing premises to see how much support they have, things rapidly get messy and complicated. There's no clear decision procedure for semisoundness. In light of this, one reasonable approach might be to focus on the easy part: get good at testing arguments for validity. Once we're sure that an argument is valid, then we can have the messy, complicated discussion about whether its premises are justified—i.e. whether the argument is semisound. Semisoundness, rather than mere validity, might be what we really want. But because validity is necessary for semisoundness (and of course soundness), and because we have a decision procedure for validity, it makes a good deal of sense to try to zero in on validity first, while setting semisoundness to one side for the time being. The tools we have are limited, but let's sharpen them to perfection.

This last point is closely tied to something I said in the prefatory note. There I said my policy would be to use silly toy examples of arguments—Harry Potter, dinosaurs, and so on—rather than real life arguments that people might actually care about. One challenge with real life arguments is that one always wants to focus on the rational acceptability of the premises. But in a way, that is getting ahead of things. Sticking with examples of arguments whose premises are just silly is a way of setting the whole issue of semisoundness to one side and focusing on the more manageable thing: validity.

Technical Terms

- Sound argument
- Semisound argument
- Formal validity requirement

§34. Abbreviated Truth Tables for Validity

|Outline|

Truth tables are a good decision procedure for validity, but they are long and cumbersome, and create opportunities for human error. Luckily, there is also a shortcut method, similar to the one used to find tautologies in §22. The trick is to think of the shortcut method as a search procedure. Is it possible for all the premises of an argument to be true, while the conclusion is false? If so, then the argument is invalid. If not, then it's valid. Let's start with some simple examples.

Here is an invalid argument:

(1) $P \vee Q$
(2) P
(3) Therefore, $\sim Q$

This one is a bit deceptive, because on the surface it looks a lot like disjunctive syllogism (§32). But it's different: in a true disjunctive syllogism, you have a premise that *negates* one of the disjuncts. Here premise (2) *affirms* one of the disjuncts. Crucially, remember that in logic, we tread '∨' as an inclusive 'or' that means something like 'and/or'. So if someone says:

(1) I'll take potions and/or defense against the dark arts.
(2) I'm taking potions.

It's still possible that they are also taking defense against the dark arts.

You should have the truth table definitions of the logical operators perfectly memorized. I know I've said this already, but if you haven't yet memorized them, stop reading immediately and go do that.

Okay, here's how to do a shortcut truth table. To begin with, just write out the argument, premise, premise, conclusion:

$P \vee Q$	P	$\sim Q$

Next, start out with the supposition that the argument is invalid, and see if you can make that work:

$P \vee Q$	P	$\sim Q$
T	**T**	**F**

Note that you place the truth value under the main operator for each statement. So for the conclusion, the 'F' goes under the tilde. From here on out, were basically just filling in Ts and Fs—in essence, reconstructing one possible line of the truth table. If it all works out, then we will have found a line on which all the premises are true while the conclusion is false, rendering the argument invalid. It's pretty easy to fill in the truth values from here: If ‘$\sim Q$’ is false, then ‘Q’ has to be true. We also know that ‘P’ is true. Once you know the truth value of a letter, you can fill that in across the board:

$P \vee Q$	P	$\sim Q$
T **T** T	**T**	**F** T

This all works out with no contradiction. So the argument is invalid. Note that what we’ve done here is merely to pick out one line of the full truth table:

premise		*conclusion*	*premise*
P	Q	$\sim Q$	$P \vee Q$
T	T	**F**	**T**
T	F	T	T
F	T	F	T
F	F	T	F

The top line, shaded in green is the one that makes the argument invalid, and it’s the same line that we zeroed in on using the shortcut truth table technique above. It’s the line where both ‘P’ and ‘Q’ are assigned values of true.

When an argument is valid, it’s not possible to have all true premises and a false conclusion. Or to put it another way, there will be no line on the truth table with all true premises and a false conclusion. So if we start out by using the shortcut method, we will in effect be searching for a line that does not exist! But that’s okay. Let’s use the shortcut method to prove the validity of disjunctive syllogism:

(1) $P \vee Q$
(2) $\sim P$
(3) Q

Again, begin by making the premises true and the conclusion false:

$P \vee Q$	$\sim P$	Q
T	**T**	**F**

This is just a supposition; we’re assigning truth values to the premises and conclusion to see if we can make this particular possibility work out. To repeat: if it *does* work out without any contradictions, then the argument is invalid. Since we know here that ‘$\sim P$’ is true, we know that ‘P’ is false. We also know that ‘Q’ is false, so we can just fill in these values for ‘P’ and ‘Q’:

$P \vee Q$	$\sim P$	Q
F **T** F	**T** F	**F**

Notice, though, that there’s a problem. The red shaded box contains a contradiction. A disjunctive statement, like ‘$P \vee Q$’, with two false disjuncts, would have to be false. But in setting up the table, we stipulated that '$P \vee Q$' is true. So this doesn't work. As a result, the argument is valid. There's no line on the truth table where it has all true premises and a false conclusion. If it helps, note that in the shortcut table above, we are basically exploring the possibility that both 'P' and 'Q' are false. If you did the full truth table for disjunctive syllogism, there would indeed be a line where both 'P' and 'Q' are false, but on that line, '$P \vee Q$' would also be false. So on that line, the argument would have a false premise.

One feature of the short truth table method that sometimes trips people up is that it seems weird to look for invalidity when you want to test to see if an argument is valid. However, validity and invalidity are exclusive and exhaustive categories. They are exclusive categories because no argument is both formally valid and formally invalid. Two categories are exclusive when there's nothing that belongs to both of them. They are also exhaustive categories because every argument is either formally valid or invalid. Generally speaking, two categories are exhaustive when there's nothing (at least, nothing from a given set of things) that belongs to neither of them. For these reasons, it's perfectly fine to test for validity by searching for invalidity first. Indeed, the definition of invalidity is the key to the whole shortcut approach: If it's *merely possible* to have all true premises and a false conclusion, then the argument is automatically invalid. So it's reasonable to look to see if that is possible.

To drive home the advantages of the shortcut method, consider the following silly example:

(1) Socrates goes to the party.
(2) Agathon goes to the party.
(3) Eryximachus goes to the party.
(4) Pausanias goes to the party.
(5) Aristophanes goes to the party.
(6) Therefore, Plato goes to the party.

This is an easy one to translate into Langerese:

(1) S
(2) G
(3) E
(4) P
(5) A
(6) Therefore, O

With six letters, a full truth table for this argument would require 64 lines. But the shortcut method makes short work of it:

S	G	E	P	A	O
T	T	T	T	T	F

Since there are no logical operators, that's all we have to do to get an answer: Yes, it is possible to have all true premises and a false conclusion. So the argument is invalid. If you wrote out the full 64 line truth table, one of those 64 lines would look exactly like this.

Unfortunately, there is one additional complication that can lead to headaches when doing shortcut truth tables. In the examples considered so far, the calculation of the truth values was relatively easy. Sometimes, though, you get stuck, because there are multiple ways of doing it. Consider:

(1) $P \vee Q$
(2) $\underline{Q \rightarrow R\qquad\qquad}$
(3) Therefore, $P \leftrightarrow R$

Start by setting this up in the usual way, assuming the premises to be true while the conclusion is false:

$P \vee Q$	$Q \rightarrow R$	$P \leftrightarrow R$
T	T	F

Here you run into trouble almost immediately. It doesn't really matter where in the table you begin, but somehow you have to assign truth values to the individual sentence letters. The problem is that there are multiple ways of doing this. If you start with the first premise, there are 3 ways of making '$P \vee Q$' true. There are also 3 ways of making '$Q \rightarrow R$' true. If you start with the conclusion, though, there are only 2 ways of making '$P \leftrightarrow R$' come out true. The bad news here is that you have explore these possibilities on multiple lines. The good news, though, is that if you start with the conclusion, you only need two lines:

$P \vee Q$	$Q \rightarrow R$	$P \leftrightarrow R$
T	T	T F F
T	T	F F T

What's going on here? Remember that we're trying to see if we can make things work out with all true premises and a false conclusion. (If so, then the argument is invalid.) To do this, the conclusion has to come out false, and with a '$\leftrightarrow$' there are only two ways of doing that. 'P' and 'R' must have opposite truth values. Since there are two possibilities here, our shortcut method won't be quite so short. But the basic approach is still the same. Try filling in each line. If things work out with no contradiction, then the argument is invalid.

$P \vee Q$	$Q \rightarrow R$	$P \leftrightarrow R$
TT	T F	T F F
T	T	F F T

Above I just filled in the truth values for ‘P’ and ‘R’ on the first line. What about ‘Q’? It turns out that ‘Q’ has to be false. If ‘Q’ were true, then ‘$Q \rightarrow R$’ would be F, since ‘R’ is false—a contradiction. So to avoid contradiction, make Q false.

$P \vee Q$	$Q \rightarrow R$	$P \leftrightarrow R$
TT F	F T F	T F F
T	T	F F T

The green shaded line works out fully, and with no contradiction. So we can say that the argument is invalid! There isn’t even any need to finish the table. This again has to do with the way in which the task is set up: we’re looking for just a single case where you have all true premises and a false conclusion, with no contradictions anywhere. Since we found that on the green line, there’s no need to look any further. On the other hand, if things had *not* worked out on the first line—if we’d hit a contradiction—then we’d have to keep going. The reason why is that it might still be possible to get all true premises and a false conclusion on the second line.

Technical Terms

- Exclusive categories
- Exhaustive categories

§35. Inductive Arguments

|Outline|

On a couple of occasions now, I've said that validity isn't necessary for having a good argument. I also defined a deductive argument (§32) as one that you construct when you are *trying* to make a valid argument. But one can also set one's sights a bit lower than validity. For example, one could *try* to make an argument where the premises, if true, would render the conclusion *probable*. When you're reasoning deductively, you're trying to make an argument where it would be impossible to have a false conclusion if all the premises are true. But one could dial things down a bit and try to make an argument where it would be *improbable* for the conclusion to be false if all the premises are true. This latter sort of argument is **inductive**.

I cannot stress enough how important it is to bear in mind that the main difference between a deductive and an inductive argument has to do with what you are *trying* to do, or what you are alleging about the relationship between the premises and the conclusion. Are you trying to show that the conclusion *would have to be true* if the premises are? Or are you trying to show that the conclusion *is probably true* if the premises are? Because this difference has to do with the arguer's goals, it may sometimes be challenging to figure out whether an argument is deductive or inductive—because sometimes people's goals just aren't very clear.

Let's say that an argument is **evidentially strong** when it actually is the case that if the premises were true, the conclusion would probably be true. Just as the goal in making a deductive argument is to make an argument that's valid, the goal in making an inductive argument is to make an argument that's strong. Validity and inductive strength sort of parallel each other. Indeed, both notions have to do with the *relationship* between the premises and the conclusion of an argument. But there are some other, really important differences between validity and strength. For one, validity is an all-or-nothing deal. An argument is either valid or invalid, full stop. But strength comes in degrees, for the simple reason that probability comes in degrees. For argument *A*, let's imagine, the probability that the conclusion is true given that the premises are true is .7. And for argument *B*, the probability that the conclusion is true given that the premises are true is .8. This means that *B* is a bit stronger than *A*, though both are strong. Of course, the numbers are just made up. In practice, it's often quite difficult to quantify inductive strength in any precise way, although probability theory and statistics can be helpful tools for thinking about inductive arguments. One other difference between validity and inductive strength is that formal validity, which we've been focusing on, is (duh) entirely a matter of form. But inductive strength is a much messier thing, having to do with both the form and the content of an argument. Finally, whereas we have an excellent decision procedure for formal validity, there just isn't any decision procedure for inductive strength.

Inductive strength is so messy and complicated that we won't dwell on it too much here. There are still many more rooms to explore in the pristine, beautiful mansions of formal logic. Nevertheless, if only to have some basis for contrast—for appreciating the simple beauty of formal logic—it's not a bad idea to get acquainted with some common patterns of inductive reasoning. For example, one sort of inductive argument that's especially common in ethics and the law is argument from analogy. First, though, a quick caveat: I'll use some letters to try to highlight the forms of some of these types of inductive arguments, but this sort of formalization is extremely sloppy compared to our development of Langerese. Previously, we've used capital letters like 'A' and 'B' to stand for statements only. Here, they might stand for any old thing.

(1) A and B both share features $F_1, F_2, F_3, \ldots$.
(2) A has feature F_n.
(3) Therefore, B probably has feature F_n.

Argument from analogy basically involves comparing two things. You note that those two things, A and B, are similar in some ways, and then you conclude that they must be similar in some other ways. The assessment of arguments by analogy can often get quite messy. You have to ask: Are there other relevant differences between A and B? How similar are A and B, really? How important are the similarities (as opposed to the differences) between them? Are they similar in ways that matter for supporting the conclusion?

A second important type of inductive argument is statistical generalization:

(1) x% of observed Fs are Gs.
(2) Therefore, x% of all Fs are Gs.

An argument such as this one involves reasoning from a sample to some conclusion about a whole set or population of things. Or you might say that statistical generalizations involve reasoning from observed cases (the sample) to unobserved cases. Again, the evaluation of statistical generalizations can get pretty messy. You have to ask: Is the sample size big enough? And is the sample representative of the broader population? For example:

(1) 15% of students at Hogwarts have at least one muggle parent.__________
(2) Therefore 15% of the entire wizarding population have at least one muggle parent.

You have to ask: Does the population of students at Hogwarts represent the larger population of witches and wizards in the right sort of way? Could the sample be biased or skewed in any way? For example, maybe families with at least one muggle parent are more likely to send their kids to Hogwarts.

Yet a third interesting type of inductive argument is inference to the best explanation:

(1) Some phenomenon, O, has been observed.
(2) $E_1, E_2, E_3, \ldots$ are all potential explanations of O.
(3) But E_2 is the best explanation from this pool.
(4) Therefore, E_2 is probably true.

Not surprisingly, the assessment of this sort of argument is also somewhat complicated. You have to ask: What exactly makes E_2 better than the other potential explanations? And how much better is it? Are there any other potential explanations that no one has thought of yet?

These are three especially common types of inductive arguments, but they aren't the only ones. There are lots of different ways in which someone could try to make an evidentially strong argument. One important thing to bear in mind is that even when arguments from analogy, statistical generalizations, and inferences to the best explanation are evidentially strong, they are still invalid! When you have an evidentially strong inductive argument, it's still possible for the conclusion to be false, even if the premise is true. That might be highly improbable—which of course is the point of saying that the argument is strong. But even highly improbable things are still possible. This is why you might occasionally hear a logician say, "All inductive arguments are invalid." That is mostly correct, though you have to add just one qualification, given how I have defined things here. It's always possible that someone could accidentally make a valid argument, when they're only trying to make an argument that's evidentially strong. Because (on the account I've given) what makes an argument inductive is the arguer's intent, this leaves open the possibility that an inductive argument might accidentally be valid.

Although we're not going to work much with inductive arguments in what follows—the focus will remain on formal deductive logic—it's worth reflecting a bit on the centrality of inductive reasoning to common sense as well as to natural science. Loads and loads of the arguments we make in ordinary life are inductive. And many arguments in scientific contexts are inductive, too. Just think for example of statistical generalizations in medical research: "Since 80% of the patients in our study who received this super drug showed improved health outcomes, the drug will probably help about 80% of people in the broader population who have this particular condition." Indeed, some philosophers have held that inductive reasoning is vastly more important than deductive reasoning—a view known as **inductivism**. The eighteenth-century Scottish philosopher David Hume is my own favorite example of a staunch inductivist. Hume is most famous, perhaps, for being skeptical about induction (and he was), but he nevertheless thought that induction is absolutely central to science and common life, and he made sure that when doing philosophy, most of his own central arguments were inductive. He worried that deductive arguments, even when perfectly crafted, were quite empty. Needless to say (since I'm writing this book), I do not really share the inductivist view. But the inductivist perspective is worth bearing in mind: formal logic has its own beauty and elegance for sure, but if you have inductivist leanings, you might wonder whether a formal language such as Langerese has any use at all when it comes to capturing the kind of reasoning that actually matters in life.

One other thing to be aware of is that in popular culture, people constantly confuse deduction and induction. For example, many people associate deduction with Sherlock Holmes. But often, when Sherlock Holmes says he's using deduction, his actual reasoning looks a lot more like inference to the best explanation.

- Argument from analogy
- Evidential strength
- Inductive argument
- Inductivism
- Inference to the Best

Technical Terms

§36. Translation and Testing

|Outline|

Unfortunately, there's no decision procedure for inductive strength. When confronted with an inductive argument, our only recourse is just to hash things out. Try to figure out how much support there is for the premises, and how well they support the conclusion. In some specialized contexts, statistical methods can help with this analysis. Often, though, the assessment of inductive arguments will just be a matter of putting our minds to work to do some critical analysis. Studying a little inductive logic can help. If you know some of the basic patterns of inductive argumentation (§35), that will help you figure out which questions to ask. It also helps to be familiar with some of the standard ways in which inductive arguments can go wrong (weak analogy, biased samples, etc.) However, having an artificial language like Langerese does not really help with the assessment of inductive arguments at all. I say this partly in order to stress the extreme limitations of formal logic. In a strange way, the limitations are related to its power. On the one hand, the tools of formal logic are quite useless for answering many of the questions that we might want to ask about arguments. But on the other hand, if we narrow our focus to validity (both the validity of arguments and what it sometimes called the formal validity of tautologous statements), then the tools of formal logic are very powerful indeed. We are now going to see how to put those tools to work.

In some earlier sections (especially §8), we looked at how to translate statements from a natural language, such as English, into Langerese. In the meantime, we've also seen how to use shortcut truth tables to test arguments for validity. Putting these two things together, we now have some extraordinarily powerful techniques at our disposal. We can take any argument under the sun, and test it for validity. There are a couple of caveats. (In logic, there is almost always a hitch of some kind.) But first let's see how the two-step procedure of translation plus testing might work in practice.

Here's an argument that somebody might make:

> If all dinosaurs had feathers, then all dinosaurs were warm-blooded. But we know that some dinosaurs did not have feathers. Therefore, not all dinosaurs were warm-blooded.

Before we translate this one, note that I've written it out in a way that makes the translation very easy. I wrote the conclusion last, and marked it with a "therefore," which makes it easy to see how to set up the argument. There are, however, gazillions of ways of stringing the same three statements together in English. For example:

> Not all dinosaurs were warm-blooded, because some dinosaurs did not have feathers. If all dinosaurs had feathers, then all dinosaurs were warm-blooded.

Or:

> Some dinosaurs did not have feathers, which means that not all dinosaurs were warm-blooded. If all dinosaurs had feathers, then all dinosaurs were warm-blooded.

Notice how the conclusion could go at the beginning of the argument, or even right in the middle. In real life, arguments are often quite a jumble. So how do you identify the conclusion? Sadly, there's no foolproof way to do that. Sometimes there are easy tip-offs, like the word "therefore." Often you just have to look at contextual cues, like the phrase "which means that" in the version just above. In general, though, the first step in any translation is to identify the conclusion. This is hugely important, because the validity of an argument often depends on which statement is the conclusion. To make this clear, consider our old friends, *modus ponens* and affirming the consequent:

$P \rightarrow Q$	$P \rightarrow Q$
P	Q
Therefore Q	Therefore P

Modus ponens, on the left, is valid. But affirming the consequent, on the right, is invalid. Notice, though that these two arguments involve the same set of statements! The only difference between them is which one is the conclusion. So when it comes to validity, correctly identifying the conclusion is sometimes the whole ballgame.

One more technical detail here deserves attention. I've said that we have a decision procedure for validity, in the form of short truth tables. That's true, but (as always) limited. Technically we only have a decision procedure for validity in Langerese. There's no decision procedure for validity in English or any other natural language. One reason for that is that validity in English is often material, rather than merely formal (§31). Another deep reason is that there's no decision procedure for translation from English into Langerese. Once we get an argument translated, we can test for validity and be sure we've got it right. But the first step in the process—the translation part—is often quite messy and mushy.

In general, when translating arguments and testing them for validity, one should adhere to the principle of charity. According to the principle of charity, when translating an argument into Langerese, one should resolve any ambiguity or unclarity in a way that makes the argument come out valid. We'll turn shortly to thinking about the rationale for adopting the principle of charity. For the moment, though, just consider how it works. Suppose you are just given three statements, without much indication of which one is the intended conclusion:

> Some dinosaurs did not have feathers. If all dinosaurs had feathers, then all dinosaurs were warm-blooded. Not all dinosaurs were warm blooded.

There are two completely different ways of ordering these statements:

$F \rightarrow W$	$F \rightarrow W$
$\sim W$	$\sim F$
Therefore, $\sim F$	Therefore, $\sim W$

Interestingly, these two reconstructions of the argument agree about the translations of each individual statement. They only disagree about which statement is the conclusion. But only the one on the left is valid—an instance of the argument pattern known as ***modus tollens***. The one on the right is invalid, a fallacious argument pattern known as **denying the antecedent**. In a case like this, where there simply aren't any clear clues about which statement is really the conclusion, the principle of charity suggests that we should err in the direction of validity, and treat this one as a *modus tollens* argument. (If you'd like, feel free to use truth tables to test the validity of these two arguments.)

Sometimes the translation of a particular statement can affect the validity of an argument. In those cases, one should also—according to the principle of charity, err in the direction of validity. But it's an interesting question how far to take this. Consider:

> Harry gets anxious when Draco Malfoy is around. Harry looks anxious. So Draco must be nearby.

The first premise of this argument is "H when D." But how should we translate that? Ordinarily you would translate this as "$D \rightarrow H$." But this makes the argument invalid—an instance of affirming the consequent:

$D \rightarrow H$
H
Therefore, D

You could make the argument valid by translating the first premise as "$H \rightarrow D$." In that case, it just becomes an instance of *modus ponens*. So the principle of charity might counsel in favor of this second translation. However, consider the difference between the following two statements:

(a) Harry gets anxious when Draco is around.
(b) Draco is around when Harry gets anxious.

It looks like "$D \rightarrow H$" is a good translation of the first statement, while "$H \rightarrow D$" is a better translation of the second. (b) is equivalent to: "When Harry gets anxious, Draco is around." Which looks like: "If Harry gets anxious, then Draco is around." (b) sounds a little strange, because it violates our intuitions about causation. In statement (a) the suggestion seems to be that Draco is a cause of Harry's anxiety. But in (b), it would be weird to suggest that Harry's anxiety is a cause of Draco's proximity! But setting that issue aside, here's the main problem: If the person making this argument had wanted to say "$H \rightarrow D$," then they probably should have said (b) rather than (a). In fact, they said (a). If you adhere to the principle of charity and translate their opening statement as "$H \rightarrow D$," you might be guilty of twisting someone else's words and attributing to them a claim that they never made!

To make this problem really vivid, consider another principle, the **principle of accurate interpretation**. According to the principle of accurate interpretation, when translating an argument into Langerese, one should always prefer the translation whose logical structure best captures what the person actually said or wrote. Interestingly, in the case we were just considering, the principle of charity and the principle of accurate interpretation might seem to pull in opposite directions. Charity suggests that we should make the argument valid by translating the first premise as "$H \rightarrow D$." But the principle of accurate interpretation suggests that we should stick with "$D \rightarrow H$," even though that makes the argument invalid.

In order to appreciate the strangeness of the principle of charity, it might be worth thinking about what happens when things go in the opposite direction. What if someone makes an argument, and it's not clear how to interpret a premise. And what if we deliberately translated the premise so as to make the argument invalid—even though an alternative translation would make it valid. Having done so, we triumphantly say: "See, so-and-so was trying to defend claim *C*, but their argument is clearly invalid!" Most logicians (and logic textbooks) regard this as a cheap argumentative trick, a logical mistake known as the **straw person fallacy**. The straw person fallacy is often a matter of uncharitable interpretation, and it involves a kind of switcheroo. When someone actually makes argument *A*, which is pretty reasonable, you instead attribute to them argument *B*, which looks a lot like *A* (and is on the same topic), but which is in fact a much weaker (possibly invalid) argument. So you attack argument *B* and then declare victory. This creates a fascinating asymmetry: if you reconstruct another person's argument in such a way as to make it seem logically worse than it is, then you are committing the straw person fallacy. But if you make the other person's argument seem logically better than it is, then you are just adhering to the principle of charity. What explains the difference?

There's nothing in the formal structure of Langerese that can really answer these questions. For example, why should the principle of charity override the principle of accurate interpretation when the two conflict? There's nothing in the syntax or semantics of Langerese that really helps here. Instead, this is (I think) a deeper question about what a formal language like Langerese is even for. It is, you might say, a question about the ethics of logic.

The principle of charity can even make a difference to whether we classify an argument as inductive or deductive. For example, as we saw in §31, many arguments in natural science have the form of affirming the consequent.

If theory *T* is true, then we should expect to make observation *O*.
We make observation *O*.
So, theory *T* is true.

For instance:

> If Darwin's evolutionary theory is true, we should expect to see transitional forms in the fossil record.
> We do in fact see transitional forms in the fossil record (like *Archaeopteryx*).
> So Darwin's theory is true.

If you construe this as a deductive argument, then it's clearly invalid, and hence falls short of the mark. But if you treat it as an inductive argument, its invalidity might make no difference. The premises might stull provide some degree of inductive support for the conclusion. Thus, the same argument might be good when treated as an inductive argument, but bad when treated as a deductive one. In such cases, the principle of charity dictates that we should treat the argument as inductive.

Technical Terms

- Principle of charity
- Principle of accurate interpretation
- Straw person fallacy

§37. Logic, Rhetoric, and the Principle of Charity

|Outline|

The world would be a vastly better place if everyone understood the difference between logic and **rhetoric**. Both are fields of study that go way back to the world of ancient Greece. Both logic and rhetoric are, in a sense, all about arguments. In fact, our earlier definition of logic as "the study of arguments" is probably too broad, because logic differs from rhetoric, and rhetoric, too, could be described as the study of arguments. So what's the difference between them? They have fundamentally different aims. The goal of rhetoric is persuasion, and it is sometimes described as "the art of persuasion." The idea is to use a variety of tricks and tactics to get people to accept a certain claim. From a rhetorical perspective, a good argument is just an argument that people in fact find convincing or persuasive. A bad argument—from a rhetorical perspective—is one that doesn't move anyone, or one whose conclusion nobody buys. Of course, this whole way of thinking about the goodness or badness of arguments is completely different from the approach we take in logic. From a logical perspective, a good argument is one that's valid (or inductively strong), and whose premises are also rationally acceptable or well-supported. A good argument, in a word, is one where the premises actually support the conclusion and give us some reason for thinking that the conclusion is true. And this is the crucial difference between logic and rhetoric: In rhetoric, persuasion is everything, and truth doesn't really matter. Thus, we might distinguish between **rhetorically good** and **logically good** arguments.

The difference between logic and rhetoric even infects how we define "argument." (This is an issue that most other logic textbooks fail to note.) Consider:

Logical definition of 'argument.' An argument is a set of statements, one or more of which are meant to *provide support* for the other, where support for a statement means reason for thinking that the statement is true.

Rhetorical definition of 'argument.' An argument is a set of statements, one or more of which are meant to *persuade* an audience to accept the other.

When we defined 'argument' at the outset, we actually built in an assumption about the goal of logic. But that's not the only way to look at arguments; indeed, the rhetorical perspective makes its own assumptions about what arguments are for.

Some logically good arguments are not especially persuasive. If human beings were ideally rational, then we would be persuaded if and only if the arguments in question were logically good. Persuasiveness and logical goodness would line up perfectly. Alas, there are many logically good arguments that fail to persuade us at all. On the other hand, many persuasive arguments are in fact logically bad.

It is difficult to overstate the importance of the distinction between logic and rhetoric. Some social and political practices and institutions privilege rhetoric over logic. When we assess arguments by seeing which ones elicit the most "likes," or which ones go viral, or which arguers receive the most votes in an election, or how much money gets spent or donated, we are basically applying rhetorical standards. We're saying the best argument is the one that's most persuasive—the one that people like, the one that gets people to spend money, the one that gets people to vote a certain way. At first glance, it might seem like persuasion is an unqualified good thing. Don't we naturally want to persuade people to join our side? However, persuasion is fundamentally about power. It is about getting people to do or think what you want them to do or think.

If your main goal is to "win" an argument (in the colloquial sense), or to bring people over to your side, or to get people to purchase your product, or to get elected to public office, or to convince people to donate to a cause, or whatever your social/political goal might be, then you will naturally have a strong motive to portray the other side's arguments as silly, unreasonable, unconvincing, or even evil. But what if, instead, you adopted a policy of not caring who "wins" the argument? What if you focused single-mindedly on figuring out which argument is the best or most reasonable one? In that case, the best approach would be to adhere to some version of the principle of charity (§36). According to the principle of charity, when reconstructing someone else's argument or attributing an argument to someone else, we should always go with the best/most reasonable version of that argument. Even if we disagree with the conclusion, we should, in a sense, work *with* the person defending that conclusion in order to see what the strongest argument for it might look like.

One thing to note about the principle of charity is that it is not a principle of logic *per se*. It is, rather, an ethical principle that involves a commitment to thinking about arguments in a logical rather than a rhetorical spirit. Rather than seeing your interlocutors as adversaries, the idea is to see them as co-inquirers, where we all have the shared project of figuring out which arguments are the best ones, logically speaking. The principle of charity also reflects a profound faith in other people. As William James once said, "faith in the fact can help create the fact." The idea here is that if we have faith in other people's rationality, even in the face of considerable evidence to the contrary, that might help enable people to meet higher logical standards of argumentation.

The principle of charity is a complex and controversial topic, and it needs much more discussion that I can give it here. One interesting question is whether it has limits, and what those limits should be. For example, there may be cases in life where someone does or says something so rotten that it makes sense to say: There is just no charitable interpretation of that. Reasonable people might disagree about where the limits of charity should be, but probably all of us can agree that there should be limits of some sort. Perhaps another issue to think about is that we often (in practice, because we are only human) apply the principle of charity inequitably, extending greater charity toward some rather than others.

In practice, the place where the principle of charity really becomes important in our study of logic is translation (see the example in §36 above). It might seem like translating arguments into Langerese and testing those arguments for validity are distinct, independent steps. However, the principle of charity complicates that when it is applied to translation. Suppose we are considering two possible translations of a natural language argument. One translation makes that argument valid, while the other makes it invalid. In this case, the principle of charity dictates that we should go with the first translation – that the first translation is the better of the two.

Technical Terms

- Logically good argument
- Rhetoric
- Rhetorically good argument

§38. Natural Deduction

|Outline|

We already have a good decision procedure for validity in the form of truth tables and especially—given human limitations—abbreviated truth tables. However, there are a couple of reasons why we might want to keep looking for other decision procedures. One reason is that later on, we'll add quantifiers to Langerese, and truth tables won't work for statements containing quantifiers. It's too early to see why that reason might be compelling, so I will just mention that here as an issue and wait to bring it into focus later on at the appropriate time. A second reason why we might want more than truth tables has to do with human psychology. Truth tables work beautifully as a decision procedure for validity, but there is also something a bit unnatural about them; they don't really track the way we actually reason from premises to conclusions when making arguments. So in this section, we'll begin to introduce a new approach—natural deduction—which is "natural" in the sense that it better tracks how we do in fact reason. Natural deduction involves the construction of proofs or **derivations**. I would also add that one advantage proofs have over truth tables is that proofs are really fun.

To start with, let's just note that we already have in hand several valid argument forms. We've met a few of these along the way:

Modus ponens $p \rightarrow q$
MP $\underline{p \qquad}$
q

Modus Tollens $p \rightarrow q$
MT $\underline{\sim q \qquad}$
$\sim p$

Disjunctive syllogism $p \vee q$ $p \vee q$
DS $\underline{\sim p \qquad}$ $\underline{\sim q \qquad}$
q p

We know that these are valid because we can establish their validity using simple truth tables. Ultimately, it's still the all-important truth tables that undergird natural deduction, at least to start with. I've listed two versions of disjunctive syllogism, where some textbooks list only one. The general idea is that disjunctive syllogism is what you do when you negate one disjunct, and conclude that the other one is true. As always, if you have any doubts, feel free to test what I'm saying by constructing truth tables for yourself. Both of those forms of disjunctive syllogism are valid. Now, natural deduction just takes advantage of these valid argument forms. Since we know that arguments fitting these patterns are always valid, we can use that to derive conclusions from premises. Suppose we're given the following argument:

1. $A \rightarrow B$

2. $\sim B$
3. $\underline{A \vee C \qquad}$
Therefore, C

If we wanted to, we could easily use a truth table or an abbreviated truth table to prove that this is valid. But there is another way. We can take advantage of the valid argument forms to construct a proof. Here's how it would look. First, write the conclusion off to the side, as follows:

1. $A \rightarrow B$
2. $\sim B$
3. $\underline{A \vee C \qquad\qquad\qquad /C \qquad}$

This shows that 'C is our target; it's the statement that we are trying to prove. The little slash is just a notational convention that is a way of saying: "This is what we want to prove." One can also use the word "show," as follows:

1. $A \rightarrow B$
2. $\sim B$
3. $\underline{A \vee C \qquad\qquad\qquad \text{Show: } C \qquad}$

The challenge is to think about how get to 'C via a series of intermediate steps or mini-arguments. Each mini-argument has to fit the pattern of one of the valid argument forms above. Here's the first step.

1. $A \rightarrow B$
2. $\sim B$
3. $\underline{A \vee C \qquad\qquad\qquad /C \qquad}$
4. $\sim A$ MT 1,2

If you look at premises 1 and 2, they have the same form as the premises for a *modus tollens* inference. If you plug them in as the premises, then you get '$\sim A$' as the conclusion. For now, just ignore line 3 of the proof above, as we don't need it yet. But look at lines 1, 2, and 4. If you put them together, they look like this:

1. $A \rightarrow B$
2. $\underline{\sim B \qquad}$
4. $\sim A$

This is a lovely substitution instance of *modus tollens*. So in the proof (as above), you write that off to the side. The 'MT' stands for *modus tollens*, and the numbers '1,2' indicate that the premises of the mini argument are lines 1 and 2. The basic idea is that each new line in the proof—in this case, line 4—is the conclusion of a mini-argument. The premises of that mini-argument have to occur somewhere earlier. Off to the right, you make a kind of note indicating what those premises are, and which valid inference pattern you are using. Because we know that *modus tollens* is valid—again that's grounded in the truth table—we know that the move from lines 1 and 2 to line 4 is a valid move. So our proof is in good shape. But what to do next?

We haven't used line 3 yet. But look at line 3 and line 4 together. Line 3 is a disjunction, and line 4 negates one of the disjuncts. So that is a nice set-up for another mini-argument: a disjunctive syllogism.

1. $A \rightarrow B$
2. $\sim B$
3. $\underline{A \vee C \qquad\qquad\qquad /C \qquad\qquad}$
4. $\sim A$ MT 1,2
5. C DS 3,4 QED

If you look at lines 3, 4, and 5 together they have exactly the same structure as disjunctive syllogism (see above). So you just note that the mini-argument has lines 3 and 4 as premises, and has the form of disjunctive syllogism (DS). Write that off to the side. But notice that line 5 is just '*C*. which is the conclusion of the original argument that we were interested in. '*C* is the statement that we were trying to prove. So, now we're done! At this point, you can stop, although it's also nice to do something to mark the fact that you've arrived at the final conclusion. There are different ways of doing this. For example, you could draw another bar above line 5. I just wrote "QED" off to the side, which stands for the Latin phrase, "*quod erat demonstrandum*," which means "what was to be shown."

A couple of other details deserve remarking upon. When listing the premises used to derive a new line, it doesn't really matter what order you write the numbers in. You could write '1,2' or '2,1'—either way is fine. Also, the ordering of the premises makes no difference. Suppose that instead of the argument above, I had given you:

1. $\sim B$
2. $A \rightarrow B$
3. $\underline{A \vee C \qquad\qquad\qquad /C}$

This is the same set of premises as the one above, but with the first two premises switched. The ordering of the premises makes no difference at all, and you can still write '$\sim A$' on line 4, deriving it from 1 and 2 via MT. Or to put the same point another way, the following two argument forms are really the same thing:

$p \rightarrow q$	$\sim q$
$\underline{\sim q \qquad\qquad\qquad}$	$\underline{p \rightarrow q \qquad\qquad}$
Therefore $\sim p$	Therefore $\sim p$

Both of these are *modus tollens*. The order in which you write the premises makes no difference.

For good measure, here's another example of a proof using the above argument forms:

1. $A \rightarrow B$
2. $\sim C$
3. $C \vee A$ $/B$

Is this a valid argument, with '*B*' as the conclusion? Turns out it is! One good strategy for doing proofs is to work backwards. If you look at line 1, you can see that if only you had '*A*' by itself, then you could derive the conclusion '*B*' via *modus ponens*. The mini-argument would look like this:

1. $A \rightarrow B$

n. A

n+1. Therefore, B MP 1,?

So you somehow need to get '*A*' on a line. But here lines 2 and 3 in the original argument can help, because they set up a disjunctive syllogism:

1. $A \rightarrow B$
2. $\sim C$
3. $C \vee A$ $/B$
4. A DS 2,3
5. B MP 1,4 QED

One advantage of proofs is that they also work easily with really long arguments involving lots of letters. For example:

1. A
2. $A \rightarrow B$
3. $B \rightarrow C$
4. $C \rightarrow D$
5. $D \rightarrow E$
6. $E \rightarrow F$
7. $F \rightarrow G$
8. $G \rightarrow H$ $/H$

Hopefully it's easy to see how to do this proof: just use *modus ponen*s over and over. Use it on lines 1 and 2 to get '*B*'. Then use that with line 3 to get '*C*', and so forth. Your proof will be a bit longer but still manageable. On the other hand, if you tried to do a full truth table, it would be ginormous.

This natural deduction technique has one severe limitation, however. If you succeed in constructing a proof, then you can be sure that the argument in question is valid—at least, you can be sure if you check each line to verify that each one follows logically from a set of premises that have already been proven. But what happens if you cannot figure out how to do a proof? What does failure mean?

There could be two reasons why you are unable to complete a proof. First, the problem might be *you*: Maybe the argument is valid, and you are just not able to see how to derive the conclusion. Second, and crucially, the problem might be *with the argument*: if the argument is invalid, then that could explain why you are unable to derive the conclusion using natural deduction. If the argument is invalid, then the conclusion does not actually follow from the premises—so of course there is no way to construct the proof. The hitch, though, is that it's hard to tell which of these two hypotheses is correct in any given case. Your only recourse is to fall back on truth tables: an abbreviated truth table can quickly tell you whether the argument is invalid, if you suspect that. This recourse, however, will not always be available; it won't work once we introduce quantifiers later on.

The formally valid argument patterns serve as inference rules for Langerese. In general, a rule of inference tells you that if you have such-and-such premises, then you can derive this or that conclusion. *Modus ponens* (MP), *modus tollens* (MT), and disjunctive syllogism (DS) are all inference rules. But these aren't the only ones. Let's now add a few more to our repertoire.

Some inference rules enable you to get rid of a logical operator. MT, MP, and DS are all like this: MP and MT get rid of the arrow, while DS gets rid of the wedge. These rules are sometimes called **exit rules**, or elimination rules. In fact, some logic textbooks use different terms for them. So instead of disjunctive syllogism (DS), some texts might tell you to write '∨E' for "wedge exit." And instead of *modus ponens* (MP) you might see '→E', for "conditional exit." But the traditional names are easy to remember and also worth knowing, so we'll stick with those here. We also need a rule for getting rid of conjunction operators, and that rule is traditionally called **simplification** (S):

Simplification $\underline{p \,\&\, q}$ $\underline{p \,\&\, q}$
S p q

This one is quite easy to remember. If you have a conjunction, 'p & q,' then you can infer either of the conjuncts. As we saw with disjunctive syllogism, we can adopt two versions of simplification. It's easy to prove both with a truth table.

So we now have exit rules for conjunction, material implication, and disjunction. What about the biconditional, '↔'? Here is a simple **biconditional exit** rule:

Biconditional Exit $\underline{p \leftrightarrow q}$ $\underline{p \leftrightarrow q}$
BE $p \rightarrow q$ $q \rightarrow p$

As before, it's easy to use a truth table to prove that these argument forms are valid. There are two versions of BE, because a biconditional statement implies two different conditional statements—hence the term 'biconditional'. A quick note: not all logic textbooks include a biconditional exit rule. That's because they rely on other rules of logical equivalence to make it possible to get rid of the biconditional operator—we'll discuss those other rules later on in §43. My own feeling about this, for what it’s worth, is that our system of logic will be most elegant if we have an elimination rule for each operator.

There is, however, one sorry exception. Alas, there’s no clear elimination rule for negation. It will turn out that the replacement rules—which are coming down the pike—will give us lots of fancy ways to get rid of and rearrange tildes. But because of the nature of the tilde, there’s no simple rule that lets you start out with a statement having the form ‘~*p*’ and gets you a conclusion with no tildes at all. This fact, perhaps, makes negation the most diabolical logical operator. You can’t get rid of it as easily as you can the others.

Natural deduction, which is the approach we’re developing here, is not actually the only way to present what you might call a system of logic. One interesting feature of natural deduction is that it does not commit us to any particular view about the status of our premises. When doing natural deduction proofs, the idea is just to treat the premises as suppositions, as in: “Supposing these premises are all true, can we show/prove that this conclusion logically follows from them?” But of course, we don’t actually know whether the premises are true at all. A much more traditional approach to logic is to try to construct an axiomatic system. The classic example of an axiomatic system is the geometry of Euclid, who lived in the late 300s and early 200s B.C.E., and which many people still study in school. Euclid’s system had five axioms, where an axiom is just a statement that is taken to have some sort of special status: Maybe the axioms are necessary truths, or true by definition, or self-evidently true. Then, in addition to the axioms, one needs a set of inference rules, just like the ones we’ve been discussing, that can be used to derive logical consequences (or theorems) from the axioms. The thought is that the special status of the axioms will then get inherited by all the theorems. If the axioms are (let’s say) self-evidently true, and if we are sure that the inference rules preserve self-evident truth, then as we derive theorems we end up with a system, the whole of which amounts to a kind of edifice of self-evidence.

The story about why many logicians today prefer a natural deduction approach is long and fascinating, with many twists and turns. The story begins, perhaps, with certain mathematicians in the 19th century who began to ask what would happen if one of Euclid’s axioms were altered. This might sound like an innocent enough question, but it turned out to have shattering consequences for the history of logic and mathematics. In the nineteenth century, a number of mathematicians, including Carl Friedrich Gauss, Nikolai Lobachevsky, and Bernhard Riemann, began exploring what would happen when Euclid’s fifth axiom (often called the “parallel postulate”) was replaced with something different. Euclid’s fifth axiom says (very roughly) that if you start with a line, and then draw two additional lines, each of which intersects the first one at a right angle, then the two new lines will never intersect each other. That is, parallel lines never intersect. The discovery that you could alter this assumption—say, by supposing that parallel lines *can* intersect—and still end up with an interesting non-Euclidean geometrical system worth studying and perhaps using, posed a deep challenge to traditional ideas about axiomatic systems. The problem had to do with the alleged special status of the axioms. If you could just change one of the axioms and thereby generate a different system of geometry, it is really difficult to sustain the claim that the axioms are all self-evident, or that they are necessary truths. There is much more to this story, but for our purposes, it is probably enough to think of natural deduction as a flexible way of doing formal logic without axioms.[2]

Technical Terms

- Axiomatic system
- Elimination (or exit) rules
- Inference rules
- Natural deduction
- Proof (or derivation)
- Valid argument forms

---

[2] For further reading on these issues in the history of logic, I strongly recommend Ernest Nagel and James R. Newman's book, *Gödel's Proof* (New York University Press, 2001).

§39. Introduction Rules

|Outline|

In the previous section, we looked at how to construct proofs, and we got acquainted with several rules that you can use to get rid of logical operators: MP, MT, DS, S, and BE. But what if you need to introduce a logical operator? We need introduction rules as well—or rules that make it possible to bring a new logical operator into a proof. Getting rid of operators is all well and good if you are trying to show the validity of an argument whose conclusion is a simple statement. But what if the conclusion itself contains an operator? For example:

1. $A \rightarrow B$
2. $\underline{A \qquad\qquad\qquad\qquad /A \& B}$

One striking thing about this argument is that there is an '&' in the conclusion but not in the premises. So somehow, you have to conjure a conjunction out of thin air. But it's possible to do this because we know—again, owing to our trusty truth tables—that there are valid argument patterns that have a conjunction in the conclusion but nowhere in the premises. The simplest of these we can call conjunction:

Conjunction	p
CON	$\underline{q \qquad}$
	$p \& q$

We can also call this rule conjunction introduction, or '&I' for short. Conjunction makes it easy to construct the above proof:

1. $A \rightarrow B$		
2. $\underline{A \qquad\qquad\qquad\qquad}$	$\underline{/A \& B}$	
3. B	MP 1,2	
4. $A \& B$	CON 2,3	QED

Now take a look at this strange argument:

1. $A \rightarrow B$
2. $\underline{A \qquad\qquad\qquad\qquad /B \vee C}$

This one might not even look valid. There is, after all, a statement letter in the conclusion that appears nowhere in the premises. Where does '*C*' come from? There is also a wedge in the conclusion, even though no wedges appear in the premises. As it happens, there is a rule, known as addition, that lets us introduce a wedge:

Addition	$\underline{p\qquad}$
ADD	$p \vee q$

We could also call this disjunction introduction, or '∨I'. This is perhaps the strangest rule we've encountered so far. To appreciate the strangeness, note that '*q*' is a variable, so we can substitute anything under the sun for '*q*'. The following, for example, is a substitution instance of ADD:

$\underline{A\qquad\qquad\qquad\qquad\qquad\qquad}$

Therefore, $A \vee [B \rightarrow (C \& D)]$

Indeed, the statement you plug in for '*p*' could be indefinitely long. Here's how to complete the above proof using the ADD rule:

1. $A \rightarrow B$	
2. $\underline{A}$	$\underline{/B \vee C}$
3. B	MP 1,2
4. $B \vee C$	ADD 3

In general, addition us a handy rule whenever you have a disjunctive conclusion. If you can prove one of the disjuncts, then you can just add the other.

We can also introduce a rule for **biconditional introduction:**

Biconditional Introduction	$p \rightarrow q$
BI	$\underline{q \rightarrow p}$
	$p \leftrightarrow q$

The BI rule lets you introduce a biconditional when you already have conditionals going both ways on previous lines. First you need the left-to-right conditional, '$p \rightarrow q$,' and then you also need the right-to-left conditional, '$q \rightarrow p$.' This makes intuitive sense, since the biconditional statement, '$p \leftrightarrow q$,' is really just a two-way conditional. To say that '*p*' and '*q*' are equivalent is to say '*p* implies *q*' and '*q* implies *p*'.

What about the arrow, though? It turns out that the rule for introducing the arrow is a bit more complicated than any of the others. Before we turn to look at that rule, let's add a couple of other fairly easy rules to our toolbox.

- Introduction rules

Technical Term

§40. Other Inference Rules: HS and CD

|Outline|

So far, we've focused on elimination rules (MP, MT, DS, BE, and S), as well as introduction rules (ADD, CON, and BI). Our discussion is a little lopsided, because we have two exit rules for '→', both MP and MT, but no introduction rules. We'll remedy that in the next section. First, though, it's worth pausing to consider a couple of other rules that are useful even if they do not function as perfect introduction or elimination rules. The first of these is **hypothetical syllogism** (HS):

Hypothetical Syllogism	$p \rightarrow q$
HS	$\underline{q \rightarrow r}$
	$p \rightarrow r$

In this case, you are not really eliminating the '→', though perhaps it could be said that you are going from two arrows to one. The HS rule is not really essential. In general, instead of using HS one could always use a combination of '→' exit (i.e., *modus ponens*) and '→' introduction rules. (The '→' introduction rule we'll look at in §41.) But HS helps to streamline things, and it also has the advantage of conforming reasonably well to our natural ways of reasoning.

The second rule worth mentioning here is one that we'll use less often, and it's one that we could also do without if we had to. And it's a bit fancier than the other inference rules. But it's pretty commonly used as a rule because it has an elegant syllogistic form. Another fun feature of this rule is that it combines three different logical operators: conditional, conjunction, and disjunction. It's called **constructive dilemma**. In general, a dilemma is a disjunctive statement. Philosophers often use the term 'dilemma' in a value laden way: a dilemma is what you have when '*A* ∨ *B*' is true, and both '*A*' and '*B*' are really bad or unpalatable options, for whatever reason. '*A*' and '*B*' are called the **horns of the dilemma**. Sometimes you will see a philosopher talking, in gruesome fashion, of someone being "impaled" on the horn of a dilemma. Being impaled on the first horn of the dilemma just means, in this case, embracing '*A*', even though that is (for whatever reason) unappealing. Sometimes what makes a horn unappealing is its logical consequences. If '*A*' is true, then perhaps '*C*' is also true—but you don't want '*C*' to be the case. Thus, it's sometimes interesting to think of dilemmas as involving conditions:

If *A* is true, then that's bad news. And if *B* is true, then that's also bad news.
Either *A* or *B*.
So either way: bad news.

Of course, instead of bad news, the news could be good! Whether the news is good or bad makes no difference to the form of the argument. This is such an interesting form of argument that it's worth trying to capture logically:

Constructive Dilemma $(p \rightarrow r)\ \&\ (q \rightarrow s)$
CD $\underline{p \vee q \qquad\qquad}$
$r \vee s$

CD is a rule that you will probably use less often than the others, but it can be incredibly useful if you ever need to prove a disjunctive conclusion.

Here is an example of a proof in which you use both HS and CD.

1. $A \rightarrow B$
2. $A \vee G$
3. $B \rightarrow F$
4. $G \rightarrow H$
5. $\underline{F \rightarrow J \qquad\qquad /J \vee H \qquad}$

Let's start by just combining some of the premises in interesting ways. Four of our five lines have arrows, and that suggests that HS might be a good rule to start with:

1. $A \rightarrow B$
2. $A \vee G$
3. $B \rightarrow F$
4. $G \rightarrow H$
5. $\underline{F \rightarrow J \qquad\qquad /J \vee H \qquad}$
6. $A \rightarrow F$ HS 1,3

It's a little hard to see what to do next, but one approach is to start with the new line, line 6, and see if you can combine it in any interesting ways with the previous lines:

1. $A \rightarrow B$
2. $A \vee G$
3. $B \rightarrow F$
4. $G \rightarrow H$
5. $\underline{F \rightarrow J \qquad\qquad /\mathrm{J} \vee \mathrm{H} \qquad}$
6. $A \rightarrow F$ HS 1,3
7. $A \rightarrow J$ HS 5,6

At this point, it might not be easy to see how to get to the conclusion at all. But notice that the conclusion is a disjunctive statement. We only have two inference rules with disjunctive conclusions: addition and CD. In order to use addition, you would have to have '*J*' on a line of its own, but there isn't really any way to get that from your premises here. So, that leaves CD. Thus, even when moves are hard to see, you can sometimes make progress by starting with the conclusion and using the process of elimination. But in order to use CD to get to the conclusion, we have to do a set-up step:

1. $A \rightarrow B$
2. $A \vee G$
3. $B \rightarrow F$
4. $G \rightarrow H$
5. $F \rightarrow J$ $\quad$ / $J \vee H$
6. $A \rightarrow F$ $\quad$ HS 1,3
7. $A \rightarrow J$ $\quad$ HS 5,6
8. $(A \rightarrow J)$ & $(G \rightarrow H)$ $\quad$ CON 4,7

Now everything is set up beautifully. Notice that the two disjuncts in the conclusion, highlighted in purple, also form the consequents of the two conditionals in line 8—a perfect set up for CD. And if you look at line 2, the two disjuncts there, highlighted in blue, are the antecedents of the two conditionals in line 8. So:

1. $A \rightarrow B$
2. $A \vee G$
3. $B \rightarrow F$
4. $G \rightarrow H$
5. $F \rightarrow J$ $\quad$ / $J \vee H$
6. $A \rightarrow F$ $\quad$ HS 1,3
7. $A \rightarrow J$ $\quad$ HS 5,6
8. $(A \rightarrow J)$ & $(G \rightarrow H)$ $\quad$ CON 4,7
9. $J \vee H$ $\quad$ CD 2,8 $\quad$ QED

As we saw earlier (in §38) when doing proofs, it helps to think of each new line as the conclusion of a mini-argument. The rule on the right refers to the form of that mini argument, and the line numbers indicate the premises. So for example, in this case, our last move is:

$(A \rightarrow J)$ & $(G \rightarrow H)$
$A \vee G$
Therefore, $J \vee H$

A nice constructive dilemma!

It's worth mentioning, for good measure, that some logic texts also like to add a further rule: destructive dilemma:

$(p \rightarrow q)$ & $(r \rightarrow s)$
$\sim q \vee \sim s$
$\sim p \vee \sim r$

Here again we have another decision point: which inference rules, exactly, should we include in our natural deduction system, and why? Here it makes sense to try to get away with as few rules as possible—in part because that makes memorization easier! We definitely need introduction and exit rules for the logical operators (except for negation, which is a special case). We don't really need HS, or CD, or destructive dilemma. But HS is awfully handy, and CD is interesting because it captures a style of argumentation that people sometimes actually use. This makes them relatively useful when translating and testing arguments for validity. Destructive dilemma is more of a borderline case. You might use it once in a long while, but I propose that we not worry about destructive dilemma and try to get by without it. (This is just a judgment call—If you really like destructive dilemma and want to use it occasionally, that's fine.)

One other quick observation: Note that CD can be thought of as a way of doing two *modus ponens* inferences at once. Destructive dilemma is a way of doing two *modus tollens* inferences at once.

- Dilemma

Technical Term

§41. Conditional Proof

|Outline|

The discussion so far has left out something really important: What's the introduction rule for the '→'? We actually have two exit rules for conditionals: *modus ponens* and *modus tollens*. But we don't have any rules at all for introducing a conditional statement. Interestingly, there is a simple argument pattern that's always valid:

$\underline{q\qquad}$
$p \rightarrow q$

In principle, one could use this as a '→' introduction rule. The more traditional approach, however, is to use a slightly different proof technique known as conditional proof. The basic idea of conditional proof is quite intuitive. If you want to prove '$p \rightarrow q$', you start by asking what if 'p' were true? Could we then prove 'q'? If so, then we know that '$p \rightarrow q$' is true. Conditional proof involves thinking about a "what if" hypothetical scenario. Here's how it works:

1\. $C \vee B$
2\. $\underline{A \rightarrow {\sim}C \qquad\qquad /A \rightarrow B}$

In this case, the conclusion you're trying to prove is an arrow statement. And there isn't really any way to use HS to derive it, because there is only one conditional statement among the premises. But we can approach this by asking: what if 'A' were true? Of course, we don't *know* whether 'A' is true. It doesn't show up among our premises, and we haven't proven it. Nor is there any way to derive it from the premises. But we can still assume it and see what happens:

1\. $C \vee B$
2\. $\underline{A \rightarrow {\sim}C \qquad\qquad /A \rightarrow B}$
    3\. A ACP

The letters "ACP" stand for "assumption for conditional proof." On line 3, we are just saying: suppose that 'A' is true. We're not listing any line numbers because 'A' doesn't actually follow from the premises above. Also, it's important to indent line 3, because we have to signal somehow that we're just exploring a what-if scenario. Once you assume 'A', however, it's easy to do a few more steps:

1\. $C \vee B$
2\. $\underline{A \rightarrow {\sim}C \qquad\qquad /A \rightarrow B}$
    3\. A ACP

4. $\tilde{} C$ MP 2,3
5. B DS 1,4

Hopefully the MP and DS steps are pretty clear. Note also that when you're working within a conditional proof (lines 3 – 5), it is perfectly fine to use premises 1 and 2. Anything you start with, and anything you've already proven, is available for use within the conditional proof. Perhaps the most important thing to remember with conditional proof is that you always have to discharge your assumption. For example, in the above proof, you cannot just stop and say, "Okay we've proven *B*." You haven't actually proven *B*, or ˜ *C*, at all. That's because the indented part of the proof is purely hypothetical. All it shows is that *if* you assume *A*, then you can prove ˜ *C*, and *B*. But if you look at your target conclusion, in a way, this is exactly what you are trying to show: *if A*, then *B*. This is exactly what the above proof shows. If you assume the antecedent of the conditional (in yellow), and if you can prove the consequent (in green), then you know that the conditional statement is true. So here is how you discharge your assumption:

1. $C \vee B$
2. $A \rightarrow \tilde{} C$ $/A \rightarrow B$
3. A ACP
4. $\tilde{} C$ MP 2,3
5. B DS 1,4
6. $A \rightarrow B$ CP 3 - 5

Notice that when you discharge your assumption, you can remove the indentation and continue with your regular proof. This is because you actually know, at this point, that '$A \rightarrow B$' has to be true if your premises on lines 1 and 2 are true. But (to repeat!) you haven't proven ˜ *C* or *B* at all. They appear in red because they are part of a "pretend" subproof. Suppose, for example, you tried to continue the above proof as follows:

1. $C \vee B$
2. $A \rightarrow \tilde{} C$ $/A \rightarrow B$
3. A ACP
4. $\tilde{} C$ MP 2,3
5. B DS 1,4
6. $A \rightarrow B$ CP 3 - 5
7. B DS 1,4 **WRONG**

I struck out that last line, 7, because it is wrong. In general, when doing a proof, you are not allowed to use lines that occur inside a conditional proof after that conditional proof has been completed. So in this case, you are not allowed to use line 4.

One other interesting feature of conditional proof is that there could be cases where you need to construct a conditional proof inside of another conditional proof. This is pretty easy to do, as long as you keep careful track of your indentations. Here is an example:

1. $C \vee B$	
2. $\underline{D \to (A \to \sim C)}$	$\underline{/D \to (A \to B)}$

If you look at the conclusion, you've got an arrow nested inside another arrow. You can prove a conclusion like this by nesting a conditional proof inside another conditional proof.

1. $C \vee B$	
2. $\underline{D \to (A \to \sim C)}$	$\underline{/D \to (A \to B)}$
$\qquad$ 3. D	ACP
$\qquad$ 4. $A \to \sim C$	MP 2,3

Notice that in starting this off, you focus on the *main operator* of the conclusion. Start by assuming the antecedent of the conclusion, which is *D*. What you want to show is that if you assume *D*, then you can prove '$A \to B$'. But how do you prove '$A \to B$'? As it happens, you can use the same conditional proof approach that we just used above:

1. $C \vee B$	
2. $\underline{D \to (A \to \sim C)}$	$\underline{/D \to (A \to B)}$
$\qquad$ 3. D	ACP
$\qquad$ 4. $A \to \sim C$	MP 2,3
$\qquad\qquad$ 5. A	ACP
$\qquad\qquad\;$ 6. $\sim C$	MP 4,5
$\qquad\qquad\;$ 7. B	DS 1,6
$\qquad$ 5. $A \to B$	CP 5 - 7

The lines in red are a little mini conditional proof, just like the one above, that get you '$A \to B$'. But we're not quite done at this point, because we still have to discharge our original assumption up on line 3. (In general, if you do a conditional proof without discharging all of your assumptions, then it's just wrong.) So here we go:

1. $C \vee B$	
2. $\underline{D \to (A \to \sim C)}$	$\underline{/D \to (A \to B)}$
$\qquad$ 3. D	ACP
$\qquad$ 4. $A \to \sim C$	MP 2,3
$\qquad\qquad$ 5. A	ACP
$\qquad\qquad\;$ 6. $\sim C$	MP 4,5
$\qquad\qquad\;$ 7. B	DS 1,6
$\qquad$ 5. $A \to B$	CP 5 - 7
6. $D \to (A \to B)$	CP 3 - 8

Here the red and blue lines all represent hypothetical premises—things you haven't really proven. But the red lines follow logically from the assumption on line 5, and the blue ones follow logically from the assumption on line 3. The red proof is a conditional proof nested inside the blue proof, which is another conditional proof.

When doing a conditional proof, there are lots of temptations to do things the wrong way. For example, since you can assume anything you want, you might be tempted to assume the conclusion you're trying to prove:

1. $C \vee B$
2. $\underline{D \rightarrow (A \rightarrow {\sim} C) \qquad\qquad /D \rightarrow (A \rightarrow B)}$
   3. $D \rightarrow (A \rightarrow B)$ ACP

However, the problem with this approach is that you still have to discharge your assumption. It's not enough just to start a conditional proof. You also have to wrap it up. This means that the conclusion has to be a conditional statement, with line 3 as its antecedent:

1. $C \vee B$
2. $\underline{D \rightarrow (A \rightarrow {\sim} C) \qquad\qquad /D \rightarrow (A \rightarrow B)}$
   3. $D \rightarrow (A \rightarrow B)$ ACP
   .
   .
   n.

n+1. $[D \rightarrow (A \rightarrow B)] \rightarrow$ ______ CP 3-n

This isn't too helpful, however. It's not your real deductive target. What you're trying to prove is '$D \rightarrow (A \rightarrow B)$', not some other statement with that as the antecedent.

Conditional proof is a powerful technique that you can use any time you want to prove a conditional statement, or as it were, any time you wish to introduce an arrow. You might want to use it in some cases where your target conclusion is a conditional statement. But you might also use it in the middle of another proof if you need to get a conditional statement as an intermediate step and you can't see any other way to do it.

There's no substitute for practice, but just to help you master the conditional proof technique, you might wish to remember the following guidelines:

1. The conclusion of conditional proof has to be a '→' statement.

2. Start by assuming the antecedent of your target statement.

3. You can assume anything you want.

4. Never, ever, forget to discharge your assumption.

5. When working within a conditional proof, you can use any of the above lines that you've actually proven—including lines within your conditional proof.

6. But once you finish your conditional proof, you are then not allowed to re-use any lines within it. The reason for this is that you haven't really established any of those lines. They are just part of a "what if" exercise.

- Conditional proof

Technical Term

§42. Using Conditional Proof to Prove Tautologies

|Outline|

Recall from §31 that any argument whose conclusion is a tautology is valid. This means, in a sense, that a tautology follows from any set of premises whatsoever—including the empty set! One fascinating thing about conditional proof is that it gives you a way of proving a statement *from no premises at all*. How could that be? To start with, consider an example of an argument with a single premise, whose conclusion is a tautology:

1. $\underline{A \qquad\qquad\qquad\qquad / J \rightarrow (K \rightarrow J)}$

It's easy to use a truth table to show that this is valid. And since we now have introduction and exit rules for all of our logical operators, we ought to be able to prove the conclusion. But how? Just use conditional proof:

1. $\underline{A \qquad\qquad\qquad\qquad / J \rightarrow (K \rightarrow J)}$
    2. J ACP

Having assumed '*J*', we now need to aim at proving '$K \rightarrow J$'. But since this is another conditional statement, we can do this using another conditional proof—nesting the conditional proofs as described in §41.

1. $\underline{A \qquad\qquad\qquad\qquad / J \rightarrow (K \rightarrow J)}$
    2. J ACP
        3. K ACP

But what do we do at this point? Here things get just a bit tricky. Once you are working inside a conditional proof, you are allowed to use any of the other lines already established. So once we assume *K* on line 3, we are allowed to use both lines 1 and 2. Of course, line 1 is useless and irrelevant. But on line 2, we know that *J* is true! So we are allowed to say, at this point, that if *K* is true, then *J* is true. We can go ahead and discharge the assumption that we made on line 3.

1. $\underline{A \qquad\qquad\qquad\qquad / J \rightarrow (K \rightarrow J)}$
    2. J ACP
        3. K ACP
    4. $K \rightarrow J$ CP 3

It might seem odd to have a conditional proof that's only 1 line. In most cases, for sure, the conditional proof will be longer. In fact, however, this works just fine. Notice also how what we just did is related to the truth table for material implication. In general, we already know that where the consequent of a conditional statement is true, the whole conditional statement is true. This means that once you've proven something on a line—like 'J', in this case—then you can use conditional proof to bring in whatever antecedent you want. In this case, we just introduced 'K' as the antecedent. Of course, we're not done yet. Every assumption has to be discharged, and we've only discharged one of them. Here's how the complete proof looks:

1. $\underline{A}$	$\underline{/\, J \rightarrow (K \rightarrow J)}$
2. J	ACP
3. K	ACP
4. $K \rightarrow J$	CP 3
5. $J \rightarrow (K \rightarrow J)$	CP 2-4

Here the idea is that if we assume 'J', we can then prove '$K \rightarrow J$'. So therefore we know that '$J \rightarrow (K \rightarrow J)$' is true. Now for one last observation: Notice how we never used line 1 anywhere at all. Line 1 is pointless and irrelevant. We don't need it at all. Here is another way to write what amounts to the same proof, but without the extra useless premise:

1. J	ACP
2. K	ACP
3. $K \rightarrow J$	CP 2
4. $J \rightarrow (K \rightarrow J)$	CP 1-3 Q.E.D.

There are no premises at all in this case. We just proved the tautology '$J \rightarrow (K \rightarrow J)$' out of nothing, *ex nihilo*. And this illustrates an interesting general rule: Where a tautology is a conditional statement, we can always prove it from nothing using conditional proof.

The above observation leads in turn to another elegant feature of Langerese. Recall that the corresponding conditional statement of every valid argument form is a tautology. Way back in §1 we started off with a simple valid argument form, *modus ponens*. Here is one instance:

$D \rightarrow O$
$\underline{D \quad\quad}$
O

And here is the corresponding conditional:

$[(D \rightarrow O) \,\&\, D] \rightarrow O$

This of course is a tautology. And it's easy to prove using the conditional proof technique:

1. $(D \rightarrow O) \,\&\, D$	ACP

You just begin by assuming (for conditional proof) the antecedent of the conditional statement you want to prove. In this case, the antecedent happens to be the conjunction of the two premises of the modus ponens argument. Here's the rest of the proof:

1. $(D \rightarrow O) \,\&\, D$	ACP
2. $D \rightarrow O$	S 1
3. D	S 1
4. O	MP 2,3
5. $[(D \rightarrow O) \,\&\, D] \rightarrow O$	ACP 1-4

There is, however, also something strange about this. The conclusion (line 5) is the corresponding conditional for *modus ponens*. But we had to use *modus ponens* to prove it! Is this a problem? Perhaps not. In this case, we're just using the rules of inference to prove a tautology, and MP is one of those rules.

§43. Logical Equivalence and Rules of Replacement

|Outline|

So far, in our development of a natural deduction system for Langerese, we've focused on inference rules, where every inference rule is basically just a valid argument form. It turns out, however, that there is also a way of putting logical equivalence to work when doing proofs. Recall that two statements are logically equivalent when they always have the same truth value. One upshot of this is that you can replace a statement with another logically equivalent statement, and that will have no effect whatsoever on the validity of the argument. The easiest way to see this is to consider one of the simplest rules of replacement. The rule of **commutativity** (COM) says that the order in which you write a conjunction or a disjunction makes no difference:

COM $p \vee q :: q \vee p$

$p \,\&\, q :: q \,\&\, p$

Here the double colon '::' just means 'is replaceable by.' It's easy to use a truth table to prove the COM rule, as well. Note also that COM does not work on the arrow, because '$p \rightarrow q$' is definitely not equivalent to '$q \rightarrow p$.' Here is another replacement rule, **material implication** (MI):

MI $p \rightarrow q :: {\sim} p \vee q$

As with the inference rules, it is a good idea to memorize these.

There are two main differences between these replacement rules and the inference rules we've studied so far. One difference is that whereas the inference rules are unidirectional, the replacement rules are bidirectional. For example, *modus ponens* goes only one way:

$p \rightarrow q$
$\underline{p \qquad}$
q

You cannot, however reason the other way:

$\underline{q \qquad}$
$p \rightarrow q$
p

Actually, '$p \rightarrow q$' does follow from 'q', but 'p' obviously does not. Replacement rules are not like this. Both of the following argument forms, for example, are fine:

$\underline{p \rightarrow q}$ $\quad\quad$ $\underline{\sim p \vee q}$
$\sim p \vee q$ $\quad\quad$ $p \rightarrow q$

Where two statements are formally equivalent, each one follows logically from the other. Logical equivalence is a two-way street.

A second difference between inference rules and replacement rules is that you can apply a replacement rule to one part of a line. You cannot do this with inference rules. Consider:

1. $A \rightarrow (B \rightarrow C)$
2. $\underline{B}$
3. Therefore, C

If you think this is an instance of *modus ponens*, then you should go back and study *modus ponens* a bit more. The problem is that line 1 does not have the right form. The premise you need for *modus ponens*, '$B \rightarrow C$', is stuck inside the line. In fact, the above argument is invalid. By contrast, you actually can replace a part of a line with an expression that's logically equivalent to it. Here is an example:

1. $\underline{A \rightarrow (B \rightarrow C)}$
2. Therefore, $A \rightarrow (\sim B \vee C)$ MI 1

This is perfectly above board. We just replaced the expression inside the parentheses with another one that we know is logically equivalent to it, thanks to the material implication rule.

Ultimately, as with the inference rules, what undergirds our use of replacement rules is the truth table technique. It's easy to construct a truth table to show that '$p \rightarrow q$' is logically equivalent to '$\sim p \vee q$'.

There are quite a number of replacement rules that we can add to the natural deduction toolbox. Here are a few helpful ones:

ASSOC $\quad$ $(p \vee q) \vee r :: p \vee (q \vee r)$

$(p \,\&\, q) \,\&\, r :: p \,\&\, (q \,\&\, r)$

The **associativity** rule lets you move parentheses and brackets around, but it only works if you have two wedges or two ampersands.

DN $\quad$ $p :: \sim\sim p$

The **double negation** rule basically says that two tildes cancel each other out. This can be really helpful for setting up certain kinds of inferences. For example:

1. $A \rightarrow \sim B$
2. $\underline{B}$ $\quad$ $/\sim A$
3. $\sim\sim B$ $\quad$ DN 2

4. $\sim A$ MT 1,3

Note that in a case like this, the *modus tollens* argument won't work unless you add both the tildes. That's because modus tollens always involves negating the consequent of a conditional. But in this case, the consequent '$\sim B$' already has a tilde. So the negation of '$\sim B$' is '$\sim\sim B$'.

DM $\sim(p \,\&\, q) :: \sim p \vee \sim q$

$\sim(p \vee q) :: \sim p \,\&\, \sim q$

This handy rule is known as **De Morgan's rule**, named after Augustus De Morgan, a 19th century logician. Be sure to memorize this rule carefully, as you will likely use it a lot. Notice how the second version of De Morgan's rule is related to the natural language phrase, "neither … nor …" There are two equally good ways of translating neither/nor statements. "Neither *p* nor *q*" means "Not (*p* or *q*)" but it also means "Not *p* and not *q*."

Although this can be a bit confusing, since 'tautology' is already a technical term, there is also a replacement rule known as **tautology**.

TAUT $p :: p \vee p$

$p :: p \,\&\, p$

In a way, we already have half of the tautology rule at our disposal. If you have 'p' on a line you can just use addition to get '$p \vee p$'. The tautology rule lets you run this the other direction as well. If you know that "Either Harry is a wizard, or Harry is a wizard," is true then you know that Harry is a wizard.

Another extremely useful rule is **transposition**:

TRANS $p \rightarrow q :: \sim q \rightarrow \sim p$

And while we're at it, we may as well introduce a simple rule that connects biconditional with conditional statements. This rule is sometimes called **material equivalence**:

ME $p \leftrightarrow q :: (p \rightarrow q) \,\&\, (q \rightarrow p)$

$p \leftrightarrow q :: (p \,\&\, q) \vee (\sim p \,\&\, \sim q)$

Note that the first version of the ME rule is very close to the biconditional exit and introduction rules (BE and BI). Technically, if you have BE and BI, as well as conjunction and simplification, you don't really need the ME rule at all. On the other hand, some textbooks will use the ME rule while skipping the BE and BI rules. Here I've decided to err in the direction of having more rules than we need, strictly speaking. This does add just a bit to the list of rules that one needs to memorize, but it also makes doing proofs a bit easier.

The **exportation** rule allows you to convert an arrow to an ampersand in the following way:

EXP $(p \,\&\, q) \rightarrow r :: p \rightarrow (q \rightarrow r)$

And finally, there is **distribution**:

DIS $p \,\&\, (q \vee r) :: (p \,\&\, q) \vee (p \,\&\, r)$

$p \vee (q \,\&\, r) :: (p \vee q) \,\&\, (p \vee r)$

Distribution can be a challenging rule to remember, especially since it comes in two flavors. It might help to bear in mind that the operator that starts out outside the parentheses ends up inside them, and *vice versa*.

These replacement rules add immense power to our natural deduction system. In a way, you can think of a proof as a game with an objective (= deriving the conclusion). For any set of premises, there is a set of possible moves, or lines that you can derive from those premises by the application of a single rule. Even with our original inference rules, the set of possible moves was infinite, thanks to addition! (Remember that you can add anything to any line.) However, setting aside addition, the space of possible moves was often quite small. Consider:

1. $p \rightarrow q$
2. p

Without addition, there are only two possible moves here: you can conjoin the two lines, to get '$p \,\&\, (p \rightarrow q)$', or else you can use *modus ponens* to get 'q'. The replacement moves, however, open the door to an indefinitely large number of moves. Here is just a sampling!

1. $p \rightarrow q$	
2. p	
3. $\sim\sim p$	DN 2
4. $\sim\sim\sim\sim p$	DN 3
5. $\sim\sim(p \rightarrow q)$	DN 1
6. $\sim p \vee q$	MI 1
7. $\sim p \vee \sim\sim q$	DN 6
8. $\sim(p \,\&\, \sim q)$	DM 7
9. $\sim(\sim q \,\&\, \sim p)$	COM 8
10. $p \,\&\, (\sim p \vee q)$	CON 2, 6
11. $(p \,\&\, \sim p) \vee (p \,\&\, q)$	DIST 10
12. $\sim q \rightarrow \sim p$	TRANS 1
13. $\sim\sim(p \,\&\, \sim p) \vee (p \,\&\, q)$	DN 11

14. $\sim(p \,\&\, \sim p) \rightarrow (p \,\&\, q)$ MI 13
etc.

Notice that in the middle, on line 10, I used an inference rule (conjunction) to mash lines 1 and 2 together into a complex statement. But all the rest of the moves involve replacement rules.

This suggests a tip for doing proofs: If you get stuck at any point, it can sometimes help to take one of your lines—or even better, the conclusion—and just play around with it using the replacement rules. See what happens. As you transform it into something different, new moves for the proof might come into view.

The replacement rules also combine in interesting ways with the introduction and exit rules for the logical operators, including conditional proof. As you gain more experience constructing proofs, you may start to see patterns and tricks. For example, we've already seen (§41) that the conclusion of any conditional proof has to be a conditional statement. But thanks to **MI**, we know that every conditional statement is logically equivalent to a disjunction. This means that we can also use conditional proof to prove any disjunction. Suppose, for example, that you are trying to prove '$A \vee B$'. At first glance, it might be really hard to see how to do this with conditional proof. But it can be done. In order to see how, let's work backwards, carefully.

n-1. $\sim\sim A \vee B$
n. $A \vee B$ DN

In thinking things through in this way, let's suppose that the last line of the proof is line n. Why use double negation? The reason is that in order to convert a disjunction to a conditional statement using **MI**, one of the disjuncts has to be negated. Working backwards from n-1, we can use **MI** to get an arrow:

n-2. $\sim A \rightarrow B$
n-1. $\sim\sim A \vee B$ MI, *n*-2
n. $A \vee B$ DN, *n*-1

Now on line *n*-2, we have a conditional statement, and one that can easily be shown with a conditional proof. To do the conditional proof, start by assuming '$\sim A$', and then try to prove 'B'. (Of course, just how this works will depend on what premises and lines you get to start with—in this example, I've left all that out.) There are many more tricks like this. But if I listed them all out, that would be like spoiling a good movie. Proofs are more fun if you can discover some tricks for yourself as you go along.

At this point, our natural deduction system is powerful indeed. It is really more powerful than it needs to be. Putting conditional proof together with the replacement rules gives us what you might call natural deductive overkill. Whenever you use conditional proof to complete a larger proof, there is always an alternative approach that uses only the replacement rules, and not conditional proof. So the approach you take is to some degree a matter of taste. To make this vivid, consider an example:

1. $\underline{F \vee (G \& H) \qquad\qquad\qquad /{\sim} G \rightarrow F}$

There are different ways of completing this proof. First, let's use conditional proof:

1. $\underline{F \vee (G \& H)}$	$\underline{/{\sim} G \rightarrow F}$	
2. $(F \vee G) \& (F \vee H)$	DIST 1	
3. $F \vee G$	S 2	
4. ${\sim} G$	ACP	
5. F	DS 3,4	
6. ${\sim} G \rightarrow F$	CP 4-5	QED

But here is an alternative approach that does not rely on conditional proof:

1. $\underline{F \vee (G \& H)}$	$\underline{/{\sim} G \rightarrow F}$	
2. $(F \vee G) \& (F \vee H)$	DIST 1	
3. $F \vee G$	S 2	
4. $G \vee F$	COM 3	
5. ${\sim}{\sim} G \vee F$	DN 4	
6. ${\sim} G \rightarrow F$	MI 5	QED

In this case, the alternative proofs were the same length—six lines. Sometimes, one approach will yield a shorter proof than the other. In such cases, the two proofs are equally good from a logical perspective—after all, any successful proof establishes the validity of the argument! And at the end of the day, establishing validity is what this is all about. However, from an aesthetic perspective, one could argue that the shorter proof is more elegant. In general, it is a laudable aesthetic goal to try to find the shortest possible way to prove the conclusion. From a logical perspective, though, the length of the proof makes no difference whatsoever.

- Replacement rules **Technical Term**

§44. Formalizing Philosophical Arguments

|Outline|

The study of logic, like learning a foreign language or a musical instrument, involves a lot of delayed gratification. You have to master a lot of concepts, and become comfortable with a variety of different techniques, before you can even begin to see how logic might be somewhat useful. But we are now (finally) at the point where our formal language—Langerese—is going to turn out to have some practical use. Consider the following argument:

> If God is all powerful (omnipotent), then he *would have been able* to create a world with no evil in it. If God is all-loving (omnibenevolent), then he *would have wanted* to create a world with no evil in it. If evil is real, then either God wanted to create evil or he was not able to create a world without it. Evil is real. Therefore, either God is not all powerful, or he is not all loving.

This is the classical deductive version of the argument from evil. There are also inductive versions of the argument. This argument is valid. Natural deduction gives us an elegant way of showing that. To start with, assign sentence constants to each of the letters:

P: God is all powerful.
L: God is all loving.
E: Evil is real.
W: God wanted to create a world with no evil.
A: God was able to create a world with no evil.

Now we can translate the argument into Langerese:

1. $P \rightarrow A$
2. $L \rightarrow W$
3. $E \rightarrow (\sim W \vee \sim A)$
4. E $/\sim P \vee \sim L$

Here is one way to complete the proof, without using conditional proof:

1. $P \rightarrow A$
2. $L \rightarrow W$
3. $E \rightarrow (\sim W \vee \sim A)$
4. E $/\sim P \vee \sim L$
5. $\sim W \vee \sim A$ MP 3,4
6. $\sim A \rightarrow \sim P$ TRANS 1
7. $\sim W \rightarrow \sim L$ TRANS 2

8. $\sim A \vee \sim W$	COM 5
9. $(\sim A \rightarrow \sim P) \,\&\, (\sim W \rightarrow \sim L)$	CON 6,7
10. $\sim P \vee \sim L$	CD 8,9

There is also an interesting way to apply conditional proof in this case. Here's a second approach:

1. $P \rightarrow A$	
2. $L \rightarrow W$	
3. $E \rightarrow (\sim W \vee \sim A)$	
4. E	$/\sim P \vee \sim L$
5. $\sim W \vee \sim A$	MP 3,4
6. P	ACP
7. A	MP 1,6
8. $\sim\sim A$	DN 7
9. $\sim W$	DS 5,8
10. $\sim L$	MT 2,9
11. $P \rightarrow \sim L$	CP 6-10
12. $\sim P \vee \sim L$	MI 11

The first version of the proof above is perhaps slightly better (in an aesthetic sense) on account of being two lines shorter.

Logic is a useful research tool in philosophy, but only when treated with appropriate regard for its limits. Logic doesn't really solve any philosophical problems. For example, it's surely a mistake to say that we can "use logic" to prove that God does or doesn't exist. What we can do, however, is use formal logic to shed light on how various statements are related. In this case, the proof shows that the deductive argument from evil is valid. Remember: that does not automatically mean that it's a good argument! We have to ask whether the argument is sound. And some philosophers might challenge that. For example, someone might challenge (4) the claim that evil is real. Perhaps we just use the term 'evil' to refer to things we don't like. Other philosophers might challenge claim (2), or the suggestion that an all loving God would not want to create a world without evil. Perhaps that claim rests on mistaken assumptions about what God's love must be like. The important thing to see here is that the deductive argument from evil doesn't settle very much. Rather, it serves as something more like a framework for further philosophical inquiry. The formal structure of the argument gives us a common reference point, and makes it easier for philosophers with a wide variety of views to avoid talking past one another, because it provides an easy way to identify points of agreement and points of divergence.

The above remarks about the role that logic might play in philosophy are closely related to the themes of form and content. Formal logic—Langerese—just gives us a fancy way of showing how statements are related. But actual philosophical problems always involve content as well as form. There is always some substantive issue at stake, and ultimately you have to figure out what view to take on that substantive issue.

§45. *Reductio ad Absurdum*

Our exploration of the natural deduction system for Langerese is nearly complete, but there is one other proof technique that's useful, especially for proving tautologies. One common pattern of argument in philosophy and mathematics is the ***reductio ad absurdum***, or reduction to absurdity. The term 'absurdity' here doesn't refer to anything like existential absurdity. Instead, 'absurdity' here just means contradiction. The basic thought is something like this: It you start with some statement, say *p*, and derive a contradiction from it, then you know that ~*p* must be true. This suggests a new way of doing a proof: assume the opposite (or the negation) of what you want to prove, and then derive a contradiction from it. This approach is sometimes also called **indirect proof.**

In some ways, indirect proof works like conditional proof. For example, you have to start with an assumption. But in this case, the assumption is the negation of your conclusion. And the target—what you are trying to get within your indirect proof—is something of the form '*p* & ~*p*'. In order to see how this works, let's take another look at the deductive argument from evil (§44).

1. $P \rightarrow A$
2. $L \rightarrow W$
3. $E \rightarrow (\sim W \vee \sim A)$
4. E /$\sim P \vee \sim L$
5. $\sim W \vee \sim A$ MP 3,4

We may as well start with the east modus ponens step. Now to do an indirect proof, just assume the opposite of the conclusion, '$\sim P \vee \sim L$'.

1. $P \rightarrow A$
2. $L \rightarrow W$
3. $E \rightarrow (\sim W \vee \sim A)$
4. E /$\sim P \vee \sim L$
5. $\sim W \vee \sim A$ MP 3,4
   6. $\sim(\sim P \vee \sim L)$ ARP (assumption for reductio)

The idea is that line 6 is the statement that we will try to reduce to absurdity, by deriving a contradiction from it.

1. $P \rightarrow A$
2. $L \rightarrow W$
3. $E \rightarrow (\sim W \vee \sim A)$
4. E /$\sim P \vee \sim L$
5. $\sim W \vee \sim A$ MP 3,4

6. $\sim(\sim P \vee \sim L)$	ARP (assumption for reductio)
7. $\sim\sim P \& \sim\sim L$	DM 6
8. $P \& L$	DN 7
9. P	S 8
10. L	S 8
11. A	MP 9,1
12. W	MP 10,2
13. $\sim\sim W$	DN 12
14. $\sim A$	DS 13, 5
15. $A \& \sim A$	CON 11,14
16. $\sim P \vee \sim L$	RP (reductio proof) 6-15.

In line 6, we in effect say: "Let's suppose that '$\sim P \vee \sim L$' is false, and see what happens." What happens, on line 15, is that the assumption leads to an absurd (or contradictory) result. So we know the assumption is false.

This indirect proof technique is especially useful for proving tautologies, or theorems of Langerese. Like conditional proof, the reductio approach enables you to prove tautologies from no premises at all. For example:

$$p \rightarrow (q \rightarrow p)$$

Because this is a conditional statement, it lends itself to conditional proof. (You might notice that this is just the corresponding conditional for disjunctive syllogism.) But we can also prove it using the indirect approach:

1. $\sim[p \rightarrow (q \rightarrow p)]$	ARP
2. $\sim[\sim p \vee (q \rightarrow p)]$	MI 1
3. $\sim[\sim p \vee (\sim q \vee p)]$	MI 2
4. $\sim\sim p \& \sim(\sim q \vee p)$	DM 3
5.$\sim\sim p$	S 4
6. p	DN 5
7. $\sim(\sim q \vee p)$	S 4
8. $\sim\sim q \& \sim p$	DM 7
9 $\sim p$	S 8
10. $p \& \sim p$	CON 6,9
11. $p \rightarrow (q \rightarrow p)$	RP 1-10.

Line 1 is the negation of the statement we're trying to prove, while line 10 is the contradiction.

Just for good measure, here is a simple reduction proof of the law of excluded middle:

1. $\sim(p \vee \sim p)$	ARP
2. $\sim p \& \sim\sim p$	DM 2
3. $\sim p \& p$	DN 2
4. $p \vee \sim p$	RP 1-2

Any tautology under the sun can be proven using *reductio ad absurdum*.

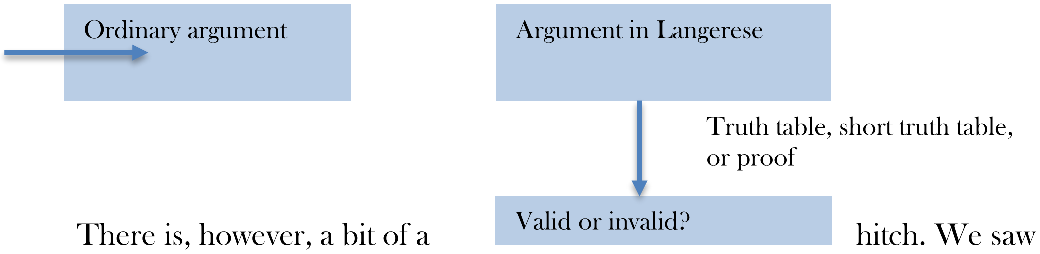
- Reductio ad absurdum

Technical Term

§46. Other Valid Arguments

|Outline|

At this point, we've come a long way. We now have multiple tools for testing arguments for formal validity. Of course, formal validity is just one ingredient of good argumentation. But we're now at the point where we can really put Langerese to work, using the following strategy:



There is, however, a bit of a hitch. We saw much earlier (§30) that there is a difference between formally valid and materially valid arguments. This distinction may yet come back to haunt us. For there are some arguments that are obviously valid, where the validity seems to have something to do with the form or structure of the argument—and yet the resources developed so far seem to fall short. So far, we have no way of capturing the validity of a wide range of arguments. Here's one example:

1. All pointer dogs are hyperactive.
2. Toby is a pointer.
3. Therefore, Toby is hyperactive.

This argument is formally valid. One way to see that is to compare it to another argument with the same form:

1. All humans are mortal.
2. Socrates is human.
3. Therefore, Socrates is mortal.

In a schematic way, we can bring out the form of these arguments as follows:

1. All Fs are G's
2. x is an F
3. Therefore, x is a G.

The problem is that there's no obvious way to translate such an argument into Langerese. If we tried, we'd have to resort to this:

1. *A*
2. *B*
3. Therefore, *C*

In a word, we'd have to treat the argument as materially valid. Here's one more example of an argument that falls through the cracks:

1. Toby is a dog who liked to snuggle.
2. Therefore, some dogs like to snuggle.

Or a bit more schematically:

1. *x* is an *F* and a *G*.
2. Therefore, some things are both *F* and *G*.

The problem is that Langerese, as we've developed it so far, has no way of capturing the validity of these and related arguments. And this is a pretty serious limitation: It turns out that lots of ordinary arguments that are obviously valid do not show up as valid if we try to translate them into Langerese. If we tried to translate the argument about Toby's penchant for snuggling, we'd get:

1. *T*
2. Therefore, *S*.

But that of course is invalid. To really drive this point home, consider another example:

1. Harry is taller than Ron.
2. Hermione is talker than Harry.
3. Therefore, Hermione is taller than Ron.

Again, this is obviously valid. But we don't have any way of showing that (yet) in Langerese. If we tried to translate it, we'd get:

1. *A*
2. *B*
3. Therefore, *C*.

What this means is that we are, in a way, just getting started. If we want to make Langerese into a really useful tool for testing arguments for validity, we have to think of ways to handle some of these other obviously valid arguments.

The basic problem is that so far, we've been treating simple statements as if they were black boxes. We've been assigning truth values to simple statements, and then using logical operators to add logical structure—for example, by forming complex statements. The problem, though, is that there is also logical structure *inside* simple statements. We might call that internal logical structure. In order to help think about this, consider that you have logical structure whenever there is some possibility of substitution. So in a complex statement, such as 'not *p*,' you can plug in any number of different things for '*p*'. But there are also possibilities for substituting things *within* a simple statement. So for example, we might say "Toby is a dog." Or "Rio is a dog." Or "Shiloh is a dog." Those look like substitution instances of the form '*x* is a dog.' And the examples just surveyed show that the internal structure of statements can sometimes make a difference to validity. The problem is that we just don't have any way of capturing that yet.

To link this issue up with other familiar concepts, note that based on everything we've said so far, we'd have to classify some of the arguments above as materially valid. This just means that we cannot establish their validity using truth tables. But this verdict is not quite right, because the validity of these arguments does seem to depend on formal structure; it's just that the formal structure that matters is inside the statements, so we have no way of symbolizing it. The solution to this problem is **predicate logic**. Predicate logic is part of classical logic, and so it helps to think of it as a way of amping up Langerese. Basically, it gives us a way of symbolizing the subject-predicate structure of arguments. It also massively expands the range of arguments that count as formally valid.

- Predicate logic

Technical Term

§47. Chrysippus's Dog

|Outline|

During the Hellenistic period, a number of rival philosophical schools sprang up in the ancient Greek speaking world. This was the time after Alexander the Great (356-323 BCE) had spread Greek culture through the middle east, and all the way to the Indus River. One group of philosophers, known as the peripatetics, were followers of Aristotle, the great philosopher who had for a while even served as tutor to Alexander. They were known as peripatetics because Aristotle himself liked to walk around town as he lectured to his students. The stoics were another important school of philosophy that was originally founded in Athens around 300 BCE. by Zeno of Citium. The group came to be known as the "stoics" because they met and talked philosophy at the "stoa," which was a particular place in Athens—a large veranda or colonnade (some would say, a porch!) where at the end of the Peloponnesian War, the "thirty" (a puppet regime installed by the victorious Spartans) had executed people. One of the greatest of the many stoic philosophers was probably Chrysippus, who lived during the 200s, BCE.

There were lots of other schools of philosophy at the time: Plato's academy still existed in Athens, and Epicureanism was also getting established. Pyrrho, a mysterious thinker who had been a soldier in the army of Alexander the Great, may well have brought a radical form of skepticism back from India. But the rivalry between the stoics and the peripatetics was especially interesting because it involved a massive disagreement about logic.

As a historical matter, the system of logic that we've been developing so far—Langerese—has its origins in ancient stoicism. (There is a sense in which all of our logic today is stoic logic.) The stoics took an interest in what we would call logical operators, and they were the first to identify valid argument forms such as *modus ponens* and disjunctive syllogism. Of all the stoic philosophers, Chrysippus was probably the one who made the most significant contribution to these efforts. (Stoics like Epictetus, Seneca, and Marcus Aurelius are more famous for their writings on ethics.)

But Aristotle was no mean slouch. Well before the stoics came on the scene, he had already, nearly singlehandedly, developed a system of logic that differed considerably from the stoical approach. And indeed, for most of western history, especially in medieval and early modern Europe, the Aristotelian approach reigned supreme. For centuries, people took Aristotle's system more or less for granted. We won't try to make a thorough study of it here, but in order to understand what comes next, it will help a great deal to know the basic idea behind Aristotle's approach.

Aristotle was deeply interested in classification. In fact, he was an enthusiastic systematic biologist and wrote extensive works on biology. When you classify something, you place it into a category. For example, you might say "birds are dinosaurs." In those cases, you're placing things in the category of dinosaurs. Aristotle also noticed that many statements combine categories:

All birds are dinosaurs – true

Some birds are dinosuars – true

No birds are dinosaurs – false

Some birds are not dinosaurs – false

These four types of statements are known as **categorical statements**, because each one of them says something about how two categories are related: the category of birds and the category of dinosaurs. Aristotle also saw that the validity or invalidity of arguments can depend on how these categories are related. For example, the following argument is valid:

All sparrows are birds.
All birds are dinosaurs.
Therefore, all sparrows are dinosaurs.

Aristotle took this insight and ran with it, developing a whole system of logic—now known as **categorical logic**—that gives a systematic account of how categorical statements can be combined and recombined to create valid (or invalid) arguments. Notice also that this argument is a lot like the ones that we looked at in §46 above: it's valid, but we don't have any way to capture that validity in the logic we've developed so far.

Now enter the stoics. If we symbolized the above argument, it would look like this:

A
B
Therefore, *C*

The stoics, led by Chrysippus, agreed that the argument is valid, but they saw it as merely materially valid. They treated it as an argument with a suppressed conditional premise. To make it formally valid, just add:

A
B
If (*A* & *B*), then *C*
Therefore, *C*

And voilá! You have an argument that stoic logic can easily handle. The peripatetic logicians, following Aristotle, thought that this move was rather silly. The argument was *already* valid because of its structure—because of the way that the categorical statements are arranged—so you don't need to add anything to it to make it formally valid.

According to legend, Chrysippus sometimes ridiculed the Aristotelian logicians by telling the following story about his hunting dog. Once, Chrysippus and his dog were tracking a rabbit down a narrow mountain path, with a wall of rock to the left and a steep drop to the right. There was nowhere for the rabbit to go but straight down the path, with the dog on its heels. But as Chrysippus rounded a bend, he witnessed the following up ahead: The rabbit was out of sight. There was a fork in the path. As his dog approached the fork, not knowing which way the rabbit went, it paused. The dog sniffed around the beginning of the left-hand fork. And then, *without sniffing any further*, the dog shot off down the right-hand path after the rabbit. Chrysippus reportedly had an "aha!" moment. "My dog is better at logic than Aristotle," he thought, "because my dog can do disjunctive syllogism. Either the rabbit went down the left-hand path or the right-hand path. It wasn't the left-hand path. So it was the right-hand path. This is a valid argument, but it's one that Aristotle's system of categorical logic has no way of capturing!" Okay, this is a little unfair to Aristotle, because he was surely also capable of disjunctive syllogism. The point is just that there are some valid argument forms not covered by Aristotle's system of categorical logic. But of course the Aristotelians thought that there are some formally valid argument forms not covered by the stoical system of propositional logic, either. And in a way, both sides of this debate were right.

In the end, what happened is that much more recently, in the nineteenth century, logicians started to work out creative ways of combining propositional logic with the Aristotelian logic of categories. These efforts led to the development of classical logic as we know it. The person who finally pulled this off, was Gottlob Frege, the German mathematician who published a little book called the *Begriffschrift* [or concept-script] in 1879. Basically, Frege worked out a way of symbolizing the categorical statements of Aristotelian logic that enabled him to combine those with the truth-functional operators of propositional logic. The notation system that Frege himself used was idiosyncratic, and nobody uses it anymore, but Frege's basic move gave rise to classical logic as we know it today. In the next several sections, we'll work our way gradually into this new synthesis of stoical and Aristotelian logic.

Technical Terms

- Categorical logic
- Categorical statement

§48 Subjects and Predicates

|Outline|

Suppose we want to symbolize a simple statement such as “Toby is a dog.” Up to now, the only option has been to give the statement a letter (or constant), say ‘*T*’. But the statement also has internal subject/predicate structure that we might want to capture. The standard way of doing this is with two letters, one for the subject and one for the predicate:

Dt

If you wanted to read this in a purely formalist way, you’d say that ‘*t*’ is a ‘*D*’. From here on out, any small italicized letter other than ‘*x*’, ‘*y*’, or ‘*z*’, will be what’s known as an **individual constant**. We need to save those three letters for use just a bit later on as statement variables. Every individual constant is letter that stands for one particular thing, or one individual. If we ever run out of letters, we can just use subscripts, so ‘t_1’ might be Toby, and ‘t_2’ might be Thor. If we want to talk about different dogs, we can just use different constants:

Dt	“Toby is a dog”
Dt_2	“Thor is a dog”
Dr	“Rio is a dog”
Ds	“Shiloh is a dog”

Etc.

Incidentally, these are all **singular statements**, which means that they say something about one singular thing. Singular statements are relatively fun and easy to translate. However, for the sake of precision, we have to say a bit more about semantics. In the above examples, the upper-case letter ‘*D*’ is serving as a **predicate constant**. It gives us a way of translating the English predicate, ‘_____ is a dog’. What makes it a constant is that it stands for the same predicate throughout.

Recall that when we were using statement constants, our semantic theory for the statement constants was really simple: The meaning of a statement constant was just some statement of the natural language that we assigned to it. (Of if you prefer, we could also say that the meaning of the statement constant is the truth value we assign to it.) But now the game has changed. An individual constant has no truth value. So what is the meaning of an individual constant? One new concept that will soon turn out to be really important is the notion of a **domain of discourse**. The domain of discourse is just the set of all the things that we might want to talk about in a particular context. To keep things really simple, we will almost always use the universal domain—the set of everything that exists! But if we wanted to, we could also narrow down the domain of discourse to the set of dogs, the set of integers, the set of philosophers, the set of Gryffyndor alumni, or the set of dinosaurs, or anything we want. We have total freedom to specify the domain of discourse in whatever way we wish. Going forward, however, just assume it's the universal domain unless I say otherwise. Now with this idea in place, we can easily give a semantic theory for individual constants: we interpret a constant by assigning it to exactly one item from the domain of discourse. There is no such thing as an empty constant that is not attached to any item from the domain. However, we could have a situation where two constants get attached to the same thing. For example, '*t*' and '*o*' could both get assigned to Toby. So far, so good.

At this point, we should pause to consider an issue that's been simmering for a while now. Some of the examples we've been considering – say, Toby and his dog friends – are things that really exist. However, other examples – say, characters from the Harry Potter books – are merely fictional people who do not really exist. There are interesting philosophical puzzles about how to understand what's going on when we reason about things that do not exist. One way to appreciate some of the puzzles is to think a bit more about the domain of discourse. Above I said that the default assumption would be that the domain of discourse is the set of everything. Should we take that to mean the set of *everything that exists*? That would mean that Harry, Hermione, and Professor McGonigall are not in the domain of discourse. Furthermore, if we stick by the rule mentioned above – i.e. that every individual constant has to be assigned to exactly one item from the domain – then there would be no way to translate statements about these characters. If we wanted to say "Professor McGonigall casts a spell," we might want to translate it like this:

Cm

But this won't really work, since '*m*' has to get assigned to something in the domain, and Professor McGonigall (since she does not exist!) is not a member of the set of everything that exists. One way to address this tricky question is to allow our domain of discourse to include fictional characters and other things that do not exist. Instead of thinking of the default domain of discourse as the set of everything that exists, we might think of it as the set of *everything that we might wish to reason or think about*, and that might include lots of fictional characters, mythological creatures, and so on. We could also specify a narrower domain that includes only fictional characters, if we wanted to. Although we can take this approach as a sort of provisional way of handling the problem of reasoning about things that do not exist, I do not think that this approach is perfect, and I encourage you to think more for yourself about how to handle this issue.

Predicates are a little bit trickier to handle, semantically, than constants. To start with, consider the simple predicate:

'________ is a dog'

What exactly does this mean? In order to appreciate the standard way of interpreting predicates, it will help a bit to think about an old-fashioned philosophical distinction between the intension and the extension of a term. The intension, roughly, is the term's meaning or sense. It is the set of properties or features that speakers ordinarily associate with the term in question. So for instance, the intension of 'dog' includes the following features:

Tail wagging
Furry
Barking
Mammal
Quadrupedal
Good sense of smell
Etc.

Of course, this is a little imprecise, and different people might have somewhat different sets of features in mind when they use the term 'dog.' For present purposes, though, don't worry too much about those details. What matters is the difference between intension and extension. The extension of the term 'dog' is a *set*: the set of all things to which the term 'dog' applies. Or you could think of it as the set of all dogs. The extension of the term 'dog' is a very, very large set:

{Toby, Mattie, Rio, Shiloh, Buddy, Skipper, Gracie, . . . }

I just started with the names of my dog and his friends here. Sometimes the extension of a term is smaller. For example, the extension of the term "eighteenth century US presidents" is just:

{Washington, Adams}

Sometimes, the extension of a term in ordinary language is not precisely delineated. In that case, we'd say that the term is vague (recall the discussion of vagueness from §4). When interpreting predicates of Langerese, we won't allow any vagueness, though. We'll assume that every predicate has a precisely delineated extension.

Interestingly, when asked to give a definition of a term, there are typically two ways of doing it. One approach is to give the intension. Another approach is to give the extension. Which of these approaches is best is a matter of huge philosophical controversy going back to Plato. If you have ever read Plato's dialogues, you will have noticed that Socrates always wants definitions: What is justice? What is piety? What is virtue? And so on. The other characters often respond by giving examples of these things—which is to say that they tend to give extensional definitions. But Socrates is never satisfied, because he always wants intensional definitions. Socrates repeatedly says: I wasn't asking for an example of piety, or justice, or whatever. I was wanting to know *what makes* pious things pious, what makes just things just, etc. Anyhow, it's also interesting to think about giving intensional *vs*. extensional definitions of logical terms. For instance, if I say that a sound argument is a valid argument having all true premises, that is an intensional definition. If I try to define soundness by pointing to examples of sound arguments and saying: "Arguments like these are the sound ones," then I am giving an extensional definition.

This was a bit of a digression, I know, but it's useful for understanding the semantic theory for predicates. The standard approach in classical logic is what we might call the extensional interpretation of predicates. If someone asks about the meaning of our predicate constant '*D*', the right response is to give the set of things from the domain of discourse to which *D* applies. This approach treats individual constants and predicate constants very similarly. In both cases, we interpret the letters by assigning them sets. The main difference is that an individual constant always gets a set containing exactly one member. A predicate could have an infinitely large extension (think about mathematical contexts where we might say something like "*x* is a real number") or it might apply to nothing at all. For example, the predicate "________ is a living non-avian dinosaur" has what we'll call an empty extension, which just means that its extension is the empty set.

To summarize what we've covered in this section so far: The domain of discourse is the set of all things we might wish to reason or talk about in a given context. When offering an interpretation of a formula such as '*Dt*', we can say that 't' is assigned to some item in our domain of discourse—say, Toby. And we can say that '*D*' is assigned a set of things from our domain of discourse—say, the set {Toby, Shiloh, Gracie, . . . }, or the set of dogs. But how then do we figure out whether '*Dt*' is true? Recall that in basic propositional logic, we'd just interpret a simple statement by assigning a truth value to it. In predicate logic, we're doing basically the same thing, but with an added twist:

> '*Dt*' is true under interpretation $\Im$ if and only if the item that $\Im$ assigns to the individual constant '*t*' belongs to the set that $\Im$ assigns to the predicate constant '*D*'.

In case it's not clear, the goofy letter $\Im$ is just a cursive 'I.' It's just a way of talking about a particular interpretation. Some logicians also talk about models instead of interpretations, but it amounts to the same thing. The above might seem like a bit much, but the basic idea is really simple, and always, it is important to be very careful about our semantic theory. Suppose that interpretation $\Im$ takes '*Dt*' to mean 'Toby is a dog.' In that case, '*Dt*' comes out true just in case Toby really does belong to the set that we assign to the predicate '*D*'—or just in case Toby really is a dog. If this sounds boring, that is good! Everything up to this point should be simple and boring.

In figuring out how to assign truth values to statements in predicate logic, we also just introduced a new way of talking when we said that a statement is true under an interpretation. Actually, the underlying concept of being true (or false) under an interpretation is one that we already worked with in propositional logic. In order to see how it works, think about an ordinary complex statement, such as '*A* & *B*'. That statement is contingent, which means (recall) that it might be true and might be false, depending on what we assign to '*A*' and '*B*'. Each way of assigning truth values to '*A*' and '*B*' – that is, each line on the truth table – is a different interpretation. So we might say that '*A* & *B*' is true under an interpretation that assigns the value of true to both '*A*' and '*B*'. But it's false under a different interpretation that specifies that '*B*' is false. The purpose of a truth table is to exhibit all the possible interpretations of a complex statement. As we develop predicate logic further, we'll continue to rely on this notion of something being true (or false) under an interpretation. It's just that we no longer have truth tables as a device to rely on.

Technical Terms

- Domain of discourse
- Extension (of a term)
- Extensional interpretation of predicates
- Individual constant
- Intension
- Predicate constant
- Truth under an interpretation

§49. Relations

|Outline|

Consider the difference between the following two statements:

Toby is a dog.

Toby loves Shiloh.

We just saw in the previous section how to symbolize the first of these two statements. But what about the second? Suppose we focus just on the predicates:

_______ is a dog.

_______ loves _______.

It's obvious that "is a dog" is what we might call a one-place predicate. Logicians also call this a **monadic predicate**. But the predicate 'loves' works differently, and relates two different things. So we might call this a relational predicate, or just a **relation**. If we wanted to symbolize "Toby loves Shiloh," we could do it like this:

Lts

And if we wanted to say that Shiloh loves Toby, we could do it as follows:

Lst

All that we did there was switch the order of the constants. If we want to say that Toby is a narcissistic dog who loves himself, we could say:

Ltt

Or if we wanted to say that Toby chases Shiloh, we could use:

Cts

Hopefully you get the idea. If we wanted to read this in a purely formalist way, without interpreting it, we might say: "*t* stands in relation *C* to *s*." One important thing about relations that these examples illustrate is that the ordering of the constants matters. If Toby's love for Shiloh is not reciprocated, then '*Lts*' might be true, while '*Lst*' might be false. This point about ordering also gives us a clue about how to interpret relations.

Remember from §48 that we interpret a monadic predicate just by assigning to it a set of items from our domain of discourse. The interpretation of a relation is very similar, except that we have to use **ordered pairs**. So let's pause to consider the difference between sets and ordered pairs. In a plain old set, the ordering of items makes no difference at all. For example:

{Plato, Aristotle}　　　　　{Aristotle, Plato}

These are really just the same set, written out in two different ways. But ordered pairs work differently:

<Plato, Aristotle>　　　　　<Aristotle, Plato>

Because the ordering of the items matters, we'd say that these are two different ordered pairs.

With this distinction between sets and ordered pairs in mind, we can see how to interpret relational predicates. We do that by assigning to the predicate a set of ordered pairs. Suppose, for example, that our predicate is "loves." We might assign to it a vast number of different ordered pairs:

____ loves _____
<Toby, Shiloh>
<me, Toby>
<Romeo, Juliet>
<Juliet, Romeo>
<me, logic>
<me, my mom>
Etc. etc. etc.

Of course, this list of ordered pairs is going to be ridiculously long! Technically, we would do better to write it out as a set:

Interpretation $\Im$ assigns to predicate *L*: {<Toby, Shiloh>, <me, Toby> . . . }

Once you see how this works, it's also pretty easy to use the same strategy we developed in §48 above to assign truth values to statements with relational predicates. We can say things like:

> '*Lts*' is true under $\Im$ if and only if the items that $\Im$ assigns to '*t*' and '*s*' form an ordered pair that belongs to the set of ordered pairs that $\Im$ assigns to *L*.

This is a bit of a mouthful, but the basic idea is still fairly simple. Suppose that interpretation $\Im$ assigns Toby to '*t*' and Shiloh to '*s*'. Then we're basically saying that '*Lts*' is true under $\Im$ if and only if the ordered pair <Toby, Shiloh> belongs to the setoff ordered pairs that $\Im$ assigns to the predicate constant *L*.

Now it might already have occurred to you that in addition to two place predicates, we might sometimes want to use three-place predicates.

_____ gave _____ to ______

_____ stole _____ from _______

Two place relations such as the ones we considered earlier are sometimes called **binary relations**. These three-place relations are a bit more complicated, but they work in the same way. For example, if we wanted to symbolize "Toby gave the frisbee to Shiloh" we could write:

Gtfs

Or if we wanted to say that Shiloh gave the frisbee to Toby, we could swap the order and say:

Gsft

We can interpret three-place relations in exactly the same way that we just interpreted binary relations, but with one difference. Instead of assigning a three-place predicate a set of ordered pairs, we have to assign it a set of ordered triples:

_____ gave _____ to _______
<Toby, the frisbee, Shiloh>
<Me, the toy, Toby>
<The person next to me on the plane, the flu, me>
Etc. etc. etc.

In practice, we will hardly ever need to worry about relations with more than three places. However, there is no principled reason why you could not have 4-place relations, or 5-place relations, or relations with 1,000 places. To interpret a 4-place relation, you'd assign it a set of ordered quadruples. To interpret a 5-place relation, you'd need a set of ordered quintuples. And so on. If you ever see a logician or mathematician talk about a set of **ordered *n*-tuples**, this is what they have in mind. The variable '*n*' just leaves it open how many places we are talking about.

Technical Terms

- Binary relation
- Monadic predicate
- Ordered n-tuple
- Ordered pair
- Relation

§50. Individual Constants and Variables

|Outline|

So far in our development of predicate logic, we've focused only on individual constants and predicate constants. We now have a good sense for how to handle the semantic theory for those. In predicate logic, though, there is an all-important distinction between individual constants and individual variables. Earlier on, when we were working through propositional logic, the difference between statement constants and statement variables never mattered all that much. Now, however, the constant/variable distinction becomes crucial, because it will have a big impact later on with respect to proofs. (You might want to read this section several times until you feel sure that you really, really understand it!)

Let's start with some notation. It is standard to use lower-case, italicized letters for both constants and variables:

Individual constants: $a, b, c, d, \ldots w, a_1, a_2, a_3, \ldots$

Individual variables: $x, y, z, x_1, x_2, x_3 \ldots$

What's the difference between a constant and a variable? The main difference is that a constant is something we interpret—something to which we give some content. Every constant has to get assigned to some item from the domain of discourse. However, **individual variables** have no content at all. They are just "dummy letters" or placeholders. You might also recall that earlier I said there can be no such thing as an empty constant that is not assigned to anything from the domain. Now perhaps you can see a little more clearly why I said that: an empty individual constant would not be a constant at all, but rather a variable. With this in mind, consider the difference between:

Dt vs. Dx

'*t* is a *D*' might be something we interpret as saying "Toby is a dog." That is an actual statement, and we assign a truth value to it in the way suggestion in §48 above. On the other hand '*x* is a *D*' is not really a statement at all. Because we are not interpreting '*x*' or assigning it to anything—it's just a placeholder—there's no way to assign a truth value to '*Dx*.' If we wanted to, we could call it a **statement function**, rather than a real statement. In general, we should be cautious about analogies between mathematics and logic. But a statement function is a little bit like a mathematical function. A mathematical function, such as 'n+1' is just an operation that can be applied to different inputs that we plug in for the variable '*n*'. Similarly, we might think of '*Dx*' as a kind of operation that can be applied to different inputs that we plug in for the individual variable '*x*.' In this case, we might also say that '*x*' is a **free variable**, meaning that it is just hanging out taking up space. None of the following formulas are actual statements, because they all contain free variables:

Dx
$Dx \,\&\, Fa$
$(Fa \,\&\, Fb) \vee Fy$
Etc.

Having just one free variable in a formula automatically demotes it from being a genuine statement to being a mere statement function. We can turn any statement function into a genuine statement with a truth value by giving it an input to work on—that is, by substituting an individual constant for the individual variable. That is a way of giving it content.

If it's impossible to have a genuine statement that contains any free variables, does that mean that a statement can never contain individual variables at all? It turns out the answer is no, because there are special cases where we can have bound variables. These special cases involve quantifiers. Recall from §47 how Aristotelian logic focused on arguments constructed out of categorical statements, such as the following:

All wizards are Hogwarts grads.
Some Hufflepuffs are Death eaters.

Here the words "all" and "some" are serving as quantifiers. It is common to call "all" the universal quantifier while "some" is the existential quantifier. Indeed, one interesting feature of Aristotelian logic is that it focuses exclusively on relationships among quantified statements. Recall that what we're trying to do now is to merge the stoical and Aristotelian traditions by combining propositional logic with predicates and quantifiers. So we'll need some symbols for the quantifiers:

$\forall x\, (Wx \rightarrow H_1x)$

This is a way of writing "all wizards are Hogwarts grads." The symbol '$\forall x$' is the universal quantifier. You might read this in the following way: For any x, if x is a wizard, then x is a Hogwarts grad. In the above quantified statement, the variable 'x' is bound by the quantifier. The expression '$Wx \rightarrow H_1x$' is a statement function, but attaching the quantifier out front turns it into a genuine statement. Similarly, consider:

$\exists x\, (H_2x \,\&\, Dx)$

Here the symbol '$\exists x$' is the existential quantifier. The above says: There is at least one x, such that x is a Hufflepuff and x is a Death eater. In case you were wondering, I used subscripts to mark that the H's are different predicates in these two cases. "Hogwarts grads" is one predicate, and "Hufflepuffs" is another. Both of these are examples of real, bona fide statements, even though they both contain variables. These statements have truth values, and well see in just a bit how to think about what would make them true or false. For the moment, though the crucial thing to see is that the variables in these statements are both bound by the quantifiers.

In order to make this into sharper focus, it might help to think about the scope of a quantifier. The scope is the portion of a statement that the quantifier governs or applies to. Consider the following examples. In each case, the scope is highlighted:

$\exists x(Fx \& Dx)$ “Something is both an *F* and a *D*”

$\exists xFx \& \exists xDx$ “Something is an *F*, and something is a *D*”

$\exists xFx \& Da$ “Something is an *F*, and a is a *D*.”

$\exists x\,(Fx \& \exists yDxy)$ “There’s something, such that it’s an *F* and it stands in relation *D* to something”

A variable is bound when it falls inside the scope of a quantifier that governs it. The variables in all of the examples above are bound, and so all of those are examples of real statements.

Sometimes, though, formulas with quantifiers can still have pesky free variables. That means that they are not genuine statements. Here are some examples:

$\exists x\mathrm{F}x \& Gx$

$\exists xDxy$

$\forall x(Fx \rightarrow Gy)$

This time, the pesky free variables are highlighted. In the first example, the problem is that the quantifier only applies to the chunk right next to it, so that the ‘*x*’ on the right is left hanging in the open. In the second and third examples, ‘*y*’ does occur in the scope of the quantifiers, but the problem is that the quantifiers only apply to ‘*x*’. They don’t govern ‘*y*.’

Why does all of this matter? It turns out that this distinction between free and bound variables is going to become very, very important when we look at techniques for constructing proofs with quantifiers. First, though, it will be helpful to do some further preliminary work. In this section, we’ve introduced some new kinds of symbols: individual variables, existential quantifiers, and universal quantifiers. And of course, every time we introduce new symbols, we also have to add a semantic theory for those symbols. What exactly do the quantifiers mean? How should we go about assigning truth values to statements with quantifiers? Unfortunately, this is the point at which semantic theory gets rather challenging. We’ll tackle the semantic theory for the quantifiers in the next section.

Technical Terms

- Bound variable
- Existential quantifier
- Free variable
- Individual variables
- Quantifier
- Scope

§51. Interpreting the Quantifiers

|Outline|

It's easy enough to say in a loose and casual way what a quantified statement might mean. Consider, for example the following argument:

All humans are mortal.
Socrates is human.
Therefore, Socrates is mortal.

This can be symbolized quire elegantly in the following way:

$\forall x\,(Hx \rightarrow Mx)$
Hs
Theferefore, Ms

This is just a lovely example of a *modus ponens* argument (§1), but one with a universal quantifier in the premise. We've already seen how to interpret 'Hs' and 'Ms' (§48). But how in the world should we interpret the first premise?

It will be helpful to begin by considering some intuitive approaches that (sadly) just don't work! It's only by trying out some things that won't work that we can begin to appreciate the rationale for the standard way of doing things. On a historical note, the standard way of interpreting the quantifiers was developed by the logician Alfred Tarski. What I will present here is a slightly casual (i.e. less-rigorous-than-usual) version of this standard approach. I think the most important thing is to begin by getting the basic idea. Then if you're interested, you can consult a more rigorous text that develops the Tarskian approach in all of its technical glory, using careful metalogical notation. In order to understand the Tarskian approach, though, it really helps to start by exploring some ideas that seem intuitive, but which turn out to be dead ends.

To begin with, our goal is to figure out how to fill in the blank in the following sort of statement:

'$\forall xFx$' is true under interpretation $\mathfrak{I}$ if and only if ___________________.

'$\exists xFx$' is true under interpretation $\mathfrak{I}$ if and only if ___________________.

At first blush, it might seem like the existential quantifier is a little bit easier to handle. Reverting to our familiar dog example, what if we said?

'$\exists xDx$' is true under interpretation $\mathfrak{I}$ if and only if 'Dt' is true under $\mathfrak{I}$.

This looks kind of promising, because we already have an account of what it means to say that 'Dt' is true under an interpretation. Alas, this approach runs into a devastating problem. If we take '$\exists xDx$' to mean, roughly, "Something is a dog," then we have to consider the possibility that our interpretation $\mathfrak{I}$ assigns the letter 't' to something that isn't actually a dog! We obviously can (if we want to) assign 't' to my dog Toby. But we could just as easily assign 't' to my next door neighbor, Terry. If our interpretation assigns 't' to Terry, then 'Dt' is *false*! – Terry is a human. But it could still be true that something is a dog – Indeed, Toby is a dog. But maybe our interpretation just assigns some *other* constant to Toby. Thus, our first try at interpreting the existential quantifier ends in failure.

In response to this failure, you might think that the problem is that we just need something a bit more complicated in the right-hand side. Maybe we just need to use logical operators. Since we already have a good semantic theory in place for the operators—truth tables—we could certainly try. Suppose our domain of discourse is the set of people present at Plato's *Symposium*—the famous drinking party where participants took turns giving speeches in praise of love:

a_1: Agathon
a_2: Aristophanes
a_3: Alcibiades
s: Socrates
p_1: Pausanias
p_2: Phaedrus
e: Eryximachus

Now suppose we wanted to say: "Everyone is Greek."

$$\forall xGx$$

Since we've restricted our domain of discourse to this very small cast of characters, that statement turns out to be true. Now, one way to say that everyone is Greek is to form a big, long conjunction:

$$Ga_1 \,\&\, Ga_2 \,\&\, Ga_3 \,\&\, Gs \,\&\, Gp_1 \,\&\, Gp_2 \,\&\, Ge$$

(Note that I left out the parentheses in order to avoid hassle. Thanks to the associativity rule, it doesn't really matter which ampersand is the main operator here.) This conjunction is called the expansion of the quantifier. Intuitively, it seems like this could help us explain what the quantifier means. In principle, even when the expansion is really long, there is a truth table that gives us the full semantic theory for it. So what if we said the following:

> '$\forall xGx$' is true under $\mathfrak{I}$ if and only if '$Ga_1 \,\&\, Ga_2 \,\&\, Ga_3 \,\&\, Gs \,\&\, Gp_1 \,\&\, Gp_2 \,\&\, Ge$' is true under $\mathfrak{I}$.

It might also occur to you that if this works, we could handle the existential quantifier in exactly the same way, but with disjunction. If we wanted to say, "At least one person from the domain is Greek," what would be equivalent to:

$Ga_1 \vee Ga_2 \vee Ga_3 \vee Gs \vee Gp_1 \vee Gp_2 \vee Ge$

We could define a universally quantified statement in terms of its conjunctive expansion, and an existentially quantified statement in terms of its disjunctive expansion.

If only. It turns out that this approach does not work well, either. The basic problem is that sometimes the domain of discourse might be infinitely large. In a way, I cheated by starting out with a finite (indeed, small) domain of discourse. But there are cases where we might wish to use an infinite domain of discourse. In fact, most of the time, our default assumption is that the domain will just be the universal set, the set of everything that we could ever want to reason about. That is surely an infinite set. But even in more specialized contexts, we run into trouble. For example, suppose we wanted to say something about "all positive integers." That is an infinitely large set, and so the expansion of our quantified statement would have to be infinitely long! And that's where we hit a technical roadblock. If you look back at §6, where we first introduced the syntax for Langerese, we saw that a wff will always be finite, because the only way to form a wff is to take two finite wffs and connect them with an operator. The problem, then, is that the grammatical rules we adopted a long time ago rule out the possibility of forming an infinitely long expansion.

At this point, you might be wondering if we are not simply up a creek. The stakes here are pretty high. If we could not actually come up with a good semantic theory for the quantifiers, then we would not be entitled to use them, and the dream of mashing together stoical and Aristotelian logic would be dashed. Luckily, Alfred Tarski came up with a way of doing this that is a little sneaky, but which actually works. Tarski's central idea is that it helps to consider two different interpretations, side-by-side. Just how this works is a little easier to see in the case of the existential quantifier. Remember the approach that failed:

> '$\exists xDx$' is true under interpretation $\Im$ if and only if 'Dt' is true under $\Im$.

And the problem was that interpretation $\Im$ might assign the letter 't' to something that isn't a dog. So it might be true under $\Im$ that *something* is a dog, but false that *t* is a dog. Now although this approach fails, it might be a little closer to success than it seemed. Suppose we have *two different interpretations*, $\Im$ and $\Im^*$. We want to figure out what's involved in saying that '$\exists xDx$' is true under $\Im$, and we might approach that by figuring out what is involved in saying that *something else* is true under $\Im^*$. That something else could just be our old friend, the statement that Toby is a dog, or 'Dt.' Here's how this might go:

> '$\exists xDx$' is true under interpretation $\Im$ if and only if 'Dt' is true under *some other* interpretation $\Im^*$ . . .

This is a start, but we need more. We have to say something about how $\Im$ and $\Im^*$ are related to each other. They have to be similar to each other! For example, if $\Im$

and ℑ* assigned completely different things to the predicate ‘*D*’, then none of this would help at all. For example, if ℑ assigns dragons to ‘*D*’ while ℑ* assigns dogs to the same predicate, then we could get an embarrassing situation where ‘*Dt*’ (or “Toby is a Dog”) is true under ℑ*, but ‘∃*xDx*’ (or “There are dragons!” is false under ℑ. What we want is to have two interpretations, ℑ and ℑ*, that are almost exactly the same, except when it comes to what they assign to ‘*t*’. In spirit, what we’re going for is this: Imagine some interpretation, ℑ*, that’s exactly like ℑ, except that ℑ* assigns the letter ‘*t*’ to something that’s really a dog. Putting it this way is a sneaky way to rule out the possibility of assigning the letter ‘*t*’ to a human, such as my next-door neighbor, Terry. This is a bit complicated, but here goes:

> ‘∃*xDx*’ is true under interpretation ℑ if and only if ‘*Dt*’ is true under *some other* interpretation ℑ*, where *the only difference* between ℑ and ℑ* is in what they assign to ‘*t*’.

It may also help to think this through backwards. Suppose we start with some interpretation ℑ* that assigns ‘*t*’ to Toby, so that ‘*Dt*’ comes out true. That’s easy enough to follow. Now imagine some other interpretation ℑ, exactly like ℑ*, except that it doesn’t assign ‘*t*’ to Toby. Maybe ℑ assigns ‘*t*’ to Terry, so that ‘*Dt*’ comes out false under ℑ. Nevertheless, we can be darned sure that if ‘*Dt*’ is true under ℑ*, and the only difference between ℑ and ℑ* is in what they assign to ‘*t*’, then it’s going to be true under ℑ that *something is a dog*.

Enough already, you are no doubt thinking. But we do still have to cover the universal quantifier. It turns out, though, that once you see how to handle the existential quantifier, we can handle the universal quantifier in the same way, without too much trouble. Here goes:

> ‘∀*xFx*’ is true under interpretation ℑ if and only if ‘Fa’ is true under *every* other interpretation ℑ*, where the only difference between ℑ and ℑ* is in what they assign to ‘*a*’.

This will serve as a semi-rigorous semantic theory for the quantifiers. As noted earlier, some other texts develop the same ideas with less preliminary discussion of dead ends, and more careful use of metalogical notation. So I invite you to think of this as a somewhat casual introduction to issues that others have covered with more technical care. One problem with my presentation here is that I’ve really only shown you how to interpret two quantified statements, ‘∃*xDx*’ and ‘∀*xFx*’, but of course there are gazillions of quantified statements. If I were being more careful, I’d state all this with greater generality, showing how to interpret any statement with the form “For any variable, predicate constant, that variable.” But hopefully this is enough to convey the general idea.

Now, if you are paying close attention, you might have noticed that there is something a little strange about this Tarskian approach to interpreting the quantifiers. In the above interpretation of the universal quantifier ‘ ∀*xFx*’, we helped ourselves to the concept of “every” on the right-hand side. Similarly, in the interpretation of '$\exists xFx$', we helped ourselves to the idea, on the right hand side, that there is some interpretation $\mathfrak{I}^*$, such that . . . This looks a little circular. It sure looks for all the world like we just used an existential quantifier ("There exists at least one interpretation $\mathfrak{I}^*$, such that …") in order to interpret the existential quantifier. Likewise for the universal quantifier. We interpret '$\forall xFx$' by talking about a statement being "true under *every* interpretation $\mathfrak{I}^*$." But that 'every' is itself a universal quantifier. So it seems that in both cases, we are using a concept to define itself, which is usually regarded as definitional dereliction. There is an adequate but superficial response to this worry. The worry arises only because we are not being careful about the distinction between the object language (i.e. Langerese) and the metalanguage (i.e. English) – see §3. What we're trying to define is the universal quantifier in Langerese, '$\forall xFx$'. In explaining how to interpret that symbol, it is perfectly fine to use whatever quantifiers we need in the metalanguage – i.e. English. So there's problem about circular definition – at least not technically.

If we push the investigation just a bit further, though, the above worry about circularity in the interpretation of the quantifiers shows up again in a deeper and more perplexing way. What is a formal language like Langerese actually for? One answer (§46) is that it's a tool for testing natural language arguments for validity. That's a good answer, but it's not the only one. Another answer is that formal logic can help us to disambiguate natural language statements. Sometimes, it's a little unclear what a natural language statement is saying, and exploring different ways of translating that statement into formal logic gives us a way of gaining precision. It has long been known that the quantifier "all" in English is ambiguous in one really important way. Consider:

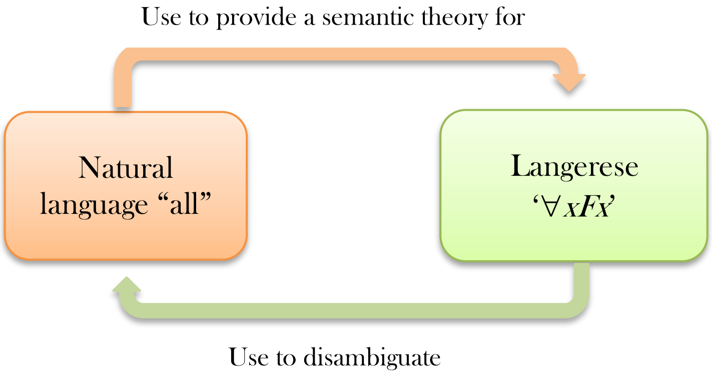
All dogs are mammals.	All dragons breathe fire.
Therefore, there are some dogs.	Therefore, there are some dragons.

This pair of arguments brings out something interesting. The one on the left seems like it might be valid. If there were no dogs, then how could all the dogs be mammals? But the one on the right seems invalid: we know that the conclusion is false, though the premise seems true. To give a name to the puzzle: it's unclear whether we should take statements like "all dogs are mammals" as having existential import. A statement has existential import when it implies the existence of the things to which its subject term refers. So "all dogs are mammals" has existential import if it implies the existence of dogs. I don't know about you, but my own intuitions about this issue are all over the map: Sometimes it *feels* like a universally quantified statement has existential import, but other times it doesn't. The point, however, is that *it is not entirely clear how to interpret the universal quantifier in English*. This is also another version of the problem, encountered earlier (§48), about how to make sense of the idea that we sometimes reason about things that do not exist.

Now, formal logic brings clarity and precision. We've already seen that the general approach is to treat statements such as "All dogs are mammals" as conditional or hypothetical statements:

All dogs are mammals.
$\forall x(Dx \rightarrow Mx)$
For any *x*, **if** *x* is a dog, then *x* is a mammal.

I placed the ‘if’ in bold type for a reason: this most emphatically does not imply that any dogs exist. The above statement could be true even if there were no dogs. So if we were using the tools of formal logic to assess the two arguments above (the ones about dogs and dragons) both would come out invalid. Thus, in Langerese, universally quantified statements never have existential import. This is yet another example of the bed of Procrustes (§8), where shades of meaning in the natural language get trimmed off and left by the wayside when we translate into Langerese. It’s a simplifying move that slices through all the natural language ambiguity of the “all” and gives us something precise that we can work with. But once you see how this works, the circularity problem might rear its ugly head again. It seems like we’re using quantifiers in the formal language to disambiguate the quantifiers in the natural language. But then we’re using quantifiers in the natural language to interpret (i.e. provide a semantic theory for) the quantifiers in the formal language.



Now, there are different ways of looking at this potential circularity. The question of how to look at this raises the deeper question of what logic is all about. (This might sound odd, but my own view is that one of the great benefits of studying logic is precisely that it positions us to ask what logic is all about.) Anyway, one possible view is that that circle is virtuous. We’re using the tools that we find (in our natural language) to help us construct new tools (Langerese) that we can then turn around and use to modify the natural language. To use a woodworking analogy: suppose you have a wooden clamp. You might use that (along with other tools) to make a better quality wood clamp. Then you might use your new clamp to hold the original one in place while you modify it, perhaps sanding off some rough edges. The tool metaphor suggests that the circularity here is not such a problem. On the other hand, there are other philosophical perspectives from which this looks suspicious. If you approach this wanting to know, “What does ‘all’ really *mean*?” you might come away disappointed. The formal language of Langerese helps us get clear about what "all" might mean in the natural language. But we can't even use the universal quantifier in the formal language without relying on the universal quantifier in the natural language when developing our semantic theory for Langerese. So it starts to feel like we're spinning in the void. This is a special case of a more general philosophical question about the relationship between a formalized language, such as Langerese, and contentful natural language.

There is one respect in which my discussion in §§48-51 has been a little bit sloppy. I have not said a peep about the grammatical or syntactical rules for predicates, individual constants or variables, or quantifiers. The syntactical rules are pretty easy, though, and there's no need to make too much of a fuss about them, as long as you keep in mind *that* as a matter of principle, we do have to have syntactical rules for every new type of symbol introduced. The rules might look a little bit like this:

> [S1] If '*a*' is an individual constant, and '*P*' is a predicate constant, then '*Pa*' is a wff.

This is just meant to show you what the syntactical rules might look like. Technically, the above rule [S1] should be formulated with metalogical variables, so that it doesn't just apply to one constant '*a*' and one predicate '*P*'. We would obviously need more rules to accommodate multi-place relations, statement functions, and quantifiers. Here's another:

> [S2] If '*Fx*' is a wff, then '∃*xFx*' is a wff.

Again, this should be formulated with metalogical variables so as to achieve generality, but the actual rule would look a bit like the above.

At this point (assuming we took the time to spell out the syntactical rules more carefully), we've reached an important milestone. Classical logic, or Langerese, is now almost fully in place. All that remains is to add on a special operator for identity (§58). The symbolic toolkit of classical logic is now almost fully accounted for, and thanks to Tarski's approach to interpreting the quantifiers, the quantifiers are legitimate. We have cut a couple of corners, syntactically. But you should now have a good idea of what both the syntax and semantics look like for Langerese with predicates, relations, and quantifiers. We are very close, now, to being able to put classical logic to work. But first, if we are going to use quantifiers in natural deduction proofs, we need some rules that will enable us both to introduce them and to get rid of them.

Technical Terms

- Existential import
- Expansion (of a quantifier)

§52. *Modus ponens* with the universal quantifier

|Outline|

The whole point of introducing predicates and quantifiers is to gain more traction in the larger project of understanding formal validity. The more logical structure we can represent symbolically, the wider the range of formally valid arguments that we can appreciate. With this in mind, consider once more the example from §51:

All humans are mortal.
Socrates is human.
Therefore, Socrates is mortal.

If we were to symbolize this in propositional logic (i.e. in Langerese without the benefit of predicates and quantifiers), it would turn out to be invalid:

P
Q
Therefore, R

But we can now symbolize it as follows:

$\forall x(Hx \rightarrow Mx)$
Hs
Therefore, Ms

This is pretty obviously an instance of *modus ponens*. But what would the actual proof look like? What's needed is some way to get rid of the quantifier. Indeed, just as we earlier had introduction and elimination rules for the (now) more familiar logical operators, we now need introduction and elimination rules for both of the quantifiers. An elimination rule for a quantifier is known as an instantiation rule. And what we need here is a rule for **universal instantiation (UI)**. The rule looks like this:

$\forall xFx$
Fa where 'a' could be any individual constant

Intuitively, this makes sense. If it's true that everything (in the domain of discourse) is an *F*, then it has to be true that *a* is an *F*. This alone is what we need to prove the above conclusion. Indeed, the proof will look like this:

1. $\forall x(Hx \rightarrow Mx)$	
2. Hs	Show: Ms
3. $Hs \rightarrow Ms$	UI 1
4. Ms	MP 2,3

In this example, line 3 is an instance of line 1. In order to form that instance, we just dropped the universal quantifier and replaced every occurrence of the variable '*x*' with the constant '*s*'. Here are some other, slightly trickier, examples of the UI rule, to help you get the idea:

$\underline{\forall x \forall y(Fx \to Gy)}$
$\forall y(Fa \to Gy)$ UI

$\underline{\forall x \forall yRxy}$
$\forall yRay$ UI

These examples bring the added complexity of having two quantifiers. In both cases, we just instantiated one of them, leaving the quantifier that's working on variable '*y*' untouched. That's why the variable '*y*' shows up in both conclusions. Note that when you use the UI rule, you can use any constant you want. So for example, the following is fine:

$\underline{\forall yRay}$
Rab UI

But we could also just as easily do this:

$\underline{\forall yRay}$
Raa UI

The first statement says that *a* stands in relation *R* to everything. If that's the case, then we can safely conclude, via UI, that *a* stands in relation *R* to *b*. But it also follows that *a* stands in relation *R* to itself.

It's also worth taking some time to consider moves that do not actually work. For example, the following argument is invalid:

$\underline{Dt \qquad\qquad\qquad\qquad}$ invalid, hence **WRONG**
Therefore, $\forall xDx$

Hopefully it's fairly easy to see why this won't work. It may be true that Toby is a dog, but it obviously does not follow from that that *everything* is a dog. It would be downright crazy to think that "everything is a *D*" follows logically from the fact that "*t* is a *D*." But once you see this, it naturally raises another question: How shall we introduce a universal quantifier? What would an introduction rule look like? This is the point at which reasoning with quantifiers gets a little complicated, and where precision is of the utmost importance.

To start with, a rule that lets you introduce a quantifier is called a generalization rule. In order to think about how this works with the universal quantifier, it will help to have another nifty example to focus on:

All dogs are mammals.
All mammals are vertebrates.
Therefore, All dogs are vertebrates.

This is the paradigm case of an old-fashioned Aristotelian categorical syllogism—a valid, two-premise argument comprised of categorical statements—and it's easy to symbolize:

$\forall x(Dx \rightarrow Mx)$
$\forall x(Mx \rightarrow Vx)$
Therefore, $\forall x(Dx \rightarrow Vx)$

This looks a lot like a hypothetical syllogism. Indeed, it *is* a hypothetical syllogism. Intuitively, the way to do the proof would be to get rid of the quantifiers in the premises, then run the hypothetical syllogism, and then bring back the universal quantifier to get the conclusion. However, if we use the UI rule as it was introduced above, we'll soon be stymied:

1. $\forall x(Dx \rightarrow Mx)$	
2. $\forall x(Mx \rightarrow Vx)$	Show: $\forall x(Dx \rightarrow Vx)$
3. $Da \rightarrow Ma$	UI 1
4. $Ma \rightarrow Va$	UI 2
5. $Da \rightarrow Va$	HS 3,4

The problem is that the inference from line 5 to the conclusion is hopelessly invalid. Going from line 5 to the conclusion would be just as bad as inferring "everything is a dog" from "Toby is a dog." This is quite annoying, because you can surely tell by relying on intuition that the Aristotelian syllogism is valid. But how to prove it?

The secret, it turns out, is to take a somewhat more permissive view of proofs, and to relax one assumption that we've been working with from the beginning. That assumption is that every line in a proof has to be a full-fledged statement. Call it the **statements only assumption**. The statements only assumption was baked into our account of natural deduction proofs from the start, where we treated each new line in a proof as the conclusion of a mini-argument. Recall the very definition of an argument as a special sort of set of statements. We're at the point, however, where we need to revisit the statements only assumption. If we stick by it, there just isn't any way to complete the proof above. But what if we dropped it, and allowed proofs with *statement functions* as well as statements? Obviously, the end result of any natural deduction proof would still have to be a full-fledged statement. The idea is that maybe, in the course of doing a proof, we could temporarily switch to using statement functions with free variables (§50). Remember how I said you should study §50 carefully and read it multiple times until you really get it? This is why.

Take another look at universal instantiation. So far, we've only used the UI rule with a constant:

$\forall xFx$ UI

Therefore, Fa

And of course, the conclusion of this inference is a statement. But if we were to drop the statements only assumption, we could also do something like the following:

$\underline{\forall xFx}$ UI

Therefore, Fy

Here the "conclusion" isn't really a conclusion, because it isn't really a statement at all. Since 'y' is a free variable, this is merely a statement form. So there are actually two different ways of doing universal instantiation: you can instantiate with a constant, or with a variable. In the above example, when applying the UI rule, I used a new variable 'y.' Technically it doesn't matter what variable you use when you instantiate via UI. But using a new variable can be a helpful way of keeping track of the fact that you've instantiated. Using a new variable also makes it a bit easier to see that the 'x' in the first statement is a bound variable, while the 'y' is free.

When constructing a proof with quantifiers, it's important to make sure that the ultimate conclusion is an actual statement—i.e. that it contains no free variables. That means that there has to be some way of getting rid of the free variables before getting to the final conclusion. Interestingly, the following move works:

$\underline{Fy}$ **Universal Generalization (UG)**

Therefore, $\forall xFx$

We saw earlier that it is not valid to generalize from a singular statement, such as "Toby is a dog" to the statement that everything is a dog. But what if, instead of a statement about Toby, we had an expression with a dummy letter that refers to no particular thing at all, as in "y is a dog," or "y is an F"? Remember that the free variable 'y' is not getting any interpretation at all; it's not assigned to anything in particular. It has no content. It's precisely because 'y' picks out nothing in particular that we can go from 'Fy' to '$\forall xFx$'. With this rule in place, let's revisit the earlier proof:

1. $\forall x(Dx \rightarrow Mx)$
2. $\underline{\forall x(Mx \rightarrow Vx)}$ Show: $\underline{\forall x(Dx \rightarrow Vx)}$
3. $Dy \rightarrow My$ UI 1
4. $My \rightarrow Vy$ UI 2

Note that here we instantiated with a variable rather than with a constant. This changes the game completely, and makes it possible to finish the proof.

5. $Dy \rightarrow Vy$ HS 3,4

This is the somewhat boring part, where we run the good old-fashioned hypothetical syllogism.

6. $\forall x(Dx \rightarrow Vx)$ UG 5

Because line 5 has the free variable 'y' rather than a constant, the UG rule works in this case.

When doing proofs with quantifiers, it can be difficult to tell at first whether one should instantiate with a constant or with a variable. Since there are two versions of the UI rule, which one should you use? Unfortunately, there is no easy answer here, for it all depends on context. However, the two cases we've covered in this section—the quantified *modus ponens* and the quantified hypothetical syllogism—afford nice examples of how you can use the two versions if UI to full effect. Basically, if your conclusion includes a quantifier, such that it looks like you might need to reintroduce that quantifier at the end using UG, then you would do well to instantiate with a variable. However, if the conclusion is about some particular thing—say, Socrates—then it might be a wiser move to instantiate with the constant that you will need in the conclusion.

Recall from earlier on that any time we adopt a new inference rule, we have to justify it. We have to be completely sure that inferences conforming to that pattern will be valid. As we saw earlier on, the easiest way to justify the adoption of an inference rule or replacement rule in propositional logic is to "back up" the rule with a truth table (§38). Now, however, things have gotten more complicated, because there is no clear way to use a truth table to "back up" the inference rules UG and UI. One simple reason for this is that we've dropped the statements only assumption. A truth table displays various combinations of truth values for statements, but you cannot construct a truth table for a mere statement form – an expression such as 'Fy' that contains a free variable. What I've tried to do, in this section, is introduce the UI and UG rules in an intuitive though informal way, so that you can see, "Oh, yeah, those inferences are going to be valid." What I have not done, however, is to back up the UG and UI rules quite as rigorously as we did earlier with the truth tables. Doing this would require constructing a more careful metalogical argument for the validity of UG and UI. I am not going to do that here – for most purposes, the intuitive understanding that (I hope) you will have gotten from this section is more than sufficient. But I also want to flag the fact that I am making a calculated decision to slack off a little bit when it comes to metalogical rigor.

There is one more technical glitch that we have to consider, and it has to do with the UG rule. There is a problem that requires a small band-aid. In order to see what the problem is, start with a statement that's obviously false:

"All dinosaurs are carnivores."

$\forall x(Dx \rightarrow Cx)$

We know this is false, because we know that some dinosaurs—like *Triceratops*—were plant-eaters. So far, so good. Now what about the following statement?

"If everyone is a dinosaur, then everyone is a carnivore"

$\forall xDx \rightarrow \forall xCx$

Is this true? It might seem counter-intuitive, but the answer is yes! Or at least, the answer is yes as long as our domain of discourse includes some things other than dinosaurs.The reason for this has to do with the truth table for conditional statements (§9). Remember that any conditional statement with a false antecedent is true. In this example, both the antecedent and the consequent are false: Not everything is a dinosaur, and not everything is a carnivore, either. But if you consult the last line in the truth table for conditional statements, it's clear that this statement comes out true:

$$\begin{array}{ccc} \forall xDx & \rightarrow & \forall xCx \\ \mathbf{F} & & \mathbf{F} \\ & \mathbf{T} & \end{array}$$

So what would happen if we tried to construct an argument out of these two statements?

> If everyone is a dinosaur, then everyone is a carnivore (True!)
> Therefore, all dinosaurs are carnivores (False!)

This argument, with a true premise and a false conclusion, is obviously invalid. But this is a problem, because it looks like we should be able to prove this conclusion, using the rules we've just added to our toolbox, along with conditional proof. Here's how the "proof" might look – I'm putting "proof" in scare quotes because there is something wrong with it:

1. $\forall xDx \rightarrow \forall xCx$	Show: $\forall x(Dx \rightarrow Cx)$	
2. Dy	ACP	
3. $\forall xDx$	UG	**WRONG**
4. $\forall xCx$	MP 2,3	
5. Cy	UI 5	
6. $Dy \rightarrow Cy$	CP 2-5	
7. $\forall x(Dx \rightarrow Cx)$	UG 6	

Obviously, this argument (as shown above) is invalid. So there is something wrong with this proof. But what? Note that lines 2 and 4 involve rules that have given us no trouble up to this point. Line 4 is just *modus ponens*. And line 2 is fine, because we you can assume anything you want when starting a conditional proof. Line 5 also seems fine, because the universal instantiation there is pretty obviously valid. If there is a problem anywhere, it must be with line 3. Even though line 3 conforms to the UG rule *as presented so far*, it's that deployment of UG inside a conditional proof that's causing trouble here. So what we need is another restriction on UG: UG may only be used in the main proof, and never in a subproof (a conditional proof or *reductio* proof). This issue may not crop up too often, but it's important to remember. Let's call this the UG restriction.

Universal generalization is especially tricky. As we saw earlier, it only works when you start out with a formula containing a free variable. And it won't work inside a conditional or an indirect proof. You should also be aware that different logic textbooks sometimes handle universal generalization in slightly different ways. One other interesting and popular approach is to introduce a whole new type of proof—universal proof—that works a little bit like conditional or indirect proof. Ultimately, it doesn't matter much whether we use the UG rule plus the UG restriction, versus a distinct universal proof technique. Both approaches have the effect of making it impossible to prove the above invalid argument.

Technical Terms

- Generalization rule
- Instantiation rule
- Statements only assumption
- UG restriction

§53. *Modus Ponens* with the existential quantifier

|Outline|

The UI and UG rules serve as tools for getting rid of and then reintroducing universal quantifiers. There are similar rules for the existential quantifier – **existential instantiation** (EI) and **existential generalization** (EG). These work a bit differently than UI and UG, and so it behooves us to take a careful look. To start with, note that there are examples of *modus ponens* that involve existential quantifiers:

All humans are mortal.
Somebody is a human.
Therefore, somebody is a mortal.

Translated into Langerese:

$\forall x(Hx \rightarrow Mx)$
$\exists xHx$
Theferefore, $\exists xMx$

Intuitively, this is a valid argument. But in order to prove the conclusion, we'll have to get rid of the quantifiers in both premises, and then (somehow) introduce the existential quantifier in the conclusion. This turns out to be something of a tricky maneuver.

To start with the easy part, introducing an existential quantifier is relatively straightforward. The following inference, for example, is valid:

Toby is a dog.	Dt	**EG**
Therefore, something is a dog.	Therefore, $\exists xDx$	

This is existential generalization. But what if we tried to go in the other direction?

Something is a dog.	$\exists xDx$
Therefore, Toby is a dog.	Therefore, Dt

This doesn't seem quite right. The problem is that we could be working with an interpretation under which the name 'Toby' has already been assigned to someone who is not a dog—say, to a person, or a cat. Suppose that 'Toby' refers to a cat. Then the premise of this inference is true—there is at least one dog!—but the conclusion is patently false. On the other hand, if this move is not allowed, then it's quite difficult to see how we could ever get rid of an existential quantifier. One possibility is to use a free variable instead of a constant:

$\exists xDx$
Dy

But this way leads to craziness, because the UG rule would then let us introduce a universal quantifier.

$\exists xDx$	Something is a dog
Dy	*y* is a dog
$\forall xDx$ (via UG)	Everything is a dog

This is absurd—the fact that there is at least one dog does not imply that everything is a dog. And the UG rule is solid (§52) so we don't want to change it. This means that the move from '$\exists xDx$' to 'Dy' doesn't work.

It turns out that we gave up on the first approach a little too quickly. Consider again:

Something is a dog.
Therefore, Toby is a dog.

This actually would be a valid inference, if we were super careful about how we are using the name 'Toby'. If we were introducing a new name—a name that has not yet been assigned to anything or anyone else—then this could work. To make this more intuitive, consider a name that (let's suppose) no one already has: Tobiwan Kenobi. Then think about it in the following way: We know that *something* is a dog. Let's take that something and *give it* the name, 'Tobiwan Kenobi.' Since no one has ever been given that name before, there's no potential for confusion. There's no chance that Tobiwan Kenobi will turn out to be a cat, or a person, because we are just insisting, by fiat, that this name is getting attached to that thing—whatever it is—that's a dog. If we can meet this new constant requirement, then the inference actually works out:

Something is a dog.
Therefore, Tobiwan Kenobi is a dog, where 'Tobiwan Kenobi' is a new name that has never been given to anything or anyone before.

Or in Langerese:

$\exists xDx$
Therefore, Dt, where 't' is a new constant that has never been assigned to anything before.

This is how existential instantiation works. We can instantiate using a constant, provided that the constant is a new one that does not yet appear anywhere else in the argument (or proof). Now we have introduction and exit rules for both the existential and the universal quantifiers. So let's put these to work.

Consider the deceptively simple *modus ponens* argument:

$\forall x(Hx \to Mx)$	"All humans are mortal."
$\exists xHx$	"Someone is human."

Theferefore, $\exists xMx$	"Therefore, someone is mortal."

I say, "deceptively simple" because the new constant requirement causes trouble right from the get-go. Suppose we commence this proof with universal instantiation:

1. $\forall x(Hx \rightarrow Mx)$	
2. $\exists xHx$	Show: $\exists xMx$
3. $Hs \rightarrow Ms$	UI 1

Now, obviously, it would be nice to get '*Hs*' on line 4, so we can do *modus ponens*, and then use existential generalization (EG) to get the conclusion. But this won't work! The trouble is that if we use existential instantiation (EI) on line 2, we have to use a new constant. If we tried to get '*Hs*' on line 4, that would violate the new constant requirement. But if we adhere to the new constant requirement, we get something that doesn't help:

1. $\forall x(Hx \rightarrow Mx)$	
2. $\exists xHx$	Show: $\exists xMx$
3. $Hs \rightarrow Ms$	UI 1
4. Ha	EI 2 + new constant

Now we're totally stymied, because we cannot use lines 3 and 4 together to do *modus ponens*. Ugh. It turns out, however, that there is a sneaky way around this problem. The way through the impasse is to do things in a different order. In commencing with this proof, we started out by getting rid of the universal quantifier first, and then moved on to the existential quantifier. But there's no reason why we have to work in that order. And if we started out with the existential quantifier, the new constant rule will not be a problem. Here goes:

1. $\forall x(Hx \rightarrow Mx)$	
2. $\exists xHx$	Show: $\exists xMx$
3. Ha	EI 2 + new constant.
4. $Ha \rightarrow Ma$	UI 1
5. Ma	MP 3,4
6. $\exists xMx$	EG 5

The moral of the story is that when your premises include both universal and existential quantifiers, it is often helpful to get rid of the existential quantifiers first. Doing things in the correct order will make it possible to complete the proof without violating the new constant rule.

As in §52, I've tried to introduce the EI and EG rules in a way that makes it intuitively clear why they are valid. But I also need to flag that I have omitted to provide a rigorous metalogical proof of their validity.

At this point, we should also pause to celebrate a bit. For Langerese is now just about complete. We have a fully worked out formal language, with logical operators, predicates, relations, and quantifiers. And although I've cut just a couple of corners (while taking care to note which corners were being cut) we also have well-worked out syntactical and semantic theories for Langerese. We also have all the inference and replacement rules that we need – above all, we have introduction and elimination roles for all of our logical operators and quantifiers. This is something of a “new car” moment, where we’ve just arrived home from the dealership with our new car. Now it’s time to put the thing through its paces and see what it can do. That is the purpose of the next few sections.

- New constant requirement

Technical Term

§54. Putting Langerese to Work

|Outline|

Consider the following philosophical argument:

> "I am only morally responsible for actions that I freely choose. If an evil hypnotist forces me to choose to do some action, then I'm not freely choosing that action. Hence, I'm not morally responsible for things that an evil hypnotist forces me to choose."

This is a simple deductive argument—the sort of argument that philosophers make all the time. We can now translate this into Langerese and prove the conclusion. The translation can be a little challenging, and it may help to walk through it step by step. Translation with predicates and quantifiers also takes a lot of practice. And there is sometimes room for disagreement about which of several alternative proposed translations is the best.

To start with, it's a good idea to start listing out the constants that you might need to use. It's okay to do this in a relatively informal way:

ℑ
i: I/me
A: ___ is an action
M: ___ is morally responsible for ___
C: ___ freely chooses ____
E: ___ is evil.
H: ___ is a hypnotist.
F: ___ forces ___ to choose _____

Often, as you list out the predicates and individual constants you'll need to use, translations of the particular statements will suggest themselves.

Start with the first line: "I am only morally responsible for actions that I freely choose." The key word here is "only." Usually, statements containing an "only" are equivalent to some other statement with a universal quantifier. For example:

> Only mammals are dogs.
> All dogs are mammals.

In this case, we can start out by rephrasing the statement in English:

> For any *x*, if *x* is an action I'm morally responsible for, then *x* is an action I freely choose.

Here is a possible translation into Langerese:

1. $\forall x[(Ax \,\&\, Mix) \rightarrow Cix]$

The second line is more complex: If an evil hypnotist forces me to choose to do some action, then I am not freely choosing that action. This statement is obviously a conditional, but it may not be clear at first glance which quantifiers to use, or how many variables we might need. One interesting feature of this statement is that the predicate "is an action" is going to appear in both the antecedent and the consequent. That suggests that we might want to use the main quantifier to focus on actions, in the following way:

> For any x, if x is an action that an evil hypnotist forces me to choose, then x is not an action that I freely choose.

With this overall structure in mind, we can start to symbolize:

$$\forall x[(Ax \,\&\, \text{an evil hypnotist forces me to choose } x) \rightarrow {\sim}(\text{I freely choose } x)]$$

One added complication here is how to translate the statement that an evil hypnotist makes me choose x. A good approach might be to use an existential quantifier: There is some y, such that y is an evil hypnotist, and y forces me to choose x. Here's how the whole statement might look:

$$2.\ \forall x[\ [Ax \,\&\, \exists y[(Ey \,\&\, Hy) \,\&\, Fyix]\ ] \rightarrow {\sim}\, Cix\ ]$$

Everything inside the red brackets falls in the scope of the main universal quantifier. Everything inside the purple brackets falls in the scope of the existential quantifier. If you read this in English, it would say:

> For any x, if x is an action, and there's some y such that y is an evil hypnotist and y forces me to choose x, then it's not the case that I choose x freely.

So far so good. The conclusion says that "I'm not morally responsible for things that an evil hypnotist forces me to choose." To work toward a good translation of this, we can reword it a bit:

> If x is an action that an evil hypnotist forces me to choose, then I'm not morally responsible for x.

Thus:

$$\forall x[[Ax \,\&\, \exists y[(Ey \,\&\, Hy) \,\&\, Fyix]] \rightarrow {\sim}Mix]$$

So here's the translation of the complete argument:

$$1.\ \forall x[(Ax \,\&\, Mix) \rightarrow Cix]$$
$$\underline{2.\ \forall x[[Ax \,\&\, \exists y[(Ey \,\&\, Hy) \,\&\, Fyix]] \rightarrow {\sim}Cix]}$$
$$\text{Therefore, } \forall x[[Ax \,\&\, \exists y[(Ey \,\&\, Hy) \,\&\, Fyix]] \rightarrow {\sim}Mix]$$

It's possible that this looks intimidating. However, the proof is actually not too bad, now that we have our introduction and exit rules for the quantifiers. We can just use UI to instantiate both of the first lines, and go from there.

Show: $\forall x[[Ax \& \exists y[(Ey \& Hy) \& Fyix]] \rightarrow \sim Mix]$

1. $\forall x[(Ax \& Mix) \rightarrow Cix]$	
~~2. $\forall x[[Ax \& \exists y[(Ey \& Hy) \& Fyix]] \rightarrow \sim Cix]$~~	
3. $(Az \& Miz) \rightarrow Ciz$	UI 1

I just used '*z*' as the instantial variable on line 3, since we already have a '*y*' in the mix. Of course, we could also instantiate line 1 with a constant, rather than a variable. However, it makes sense to use a variable because we are going to have to use the UG rule to reintroduce the universal quantifier in the conclusion.

1. $\forall x[(Ax \& Mix) \rightarrow Cix]$	
~~2. $\forall x[[Ax \& \exists y[(Ey \& Hy) \& Fyix]] \rightarrow \sim Cix]$~~	
3. $(Az \& Miz) \rightarrow Ciz$	UI 1
4. $[Az \& \exists y[(Ey \& Hy) \& Fyiz]] \rightarrow \sim Ciz$	UI 2

In using the UI rule on line 2, I re-used the same variable 'z,' because having 'z' in both lines 3 and 4 will make it possible to do some logical work on those lines. But now what? Notice that the conclusion is a conditional statement. That means that conditional proof could be a helpful tool. Because the conclusion has a universal quantifier out front, we'll need to use UG to bring in the quantifier at the very end. But suppose we could figure out how to get the following:

$[Az \& \exists y[(Ey \& Hy) \& Fyiz] \rightarrow \sim Miz$

Then all we'd have to do is use UG to get the conclusion. But since the above statement is a conditional statement, we can get it via conditional proof. Here's how to set it up:

1. $\forall x[(Ax \& Mix) \rightarrow Cix]$	
~~2. $\forall x[[Ax \& \exists y[(Ey \& Hy) \& Fyix]] \rightarrow \sim Cix]$~~	
3. $(Az \& Miz) \rightarrow Ciz$	UI 1
4. $[Az \& \exists y[(Ey \& Hy) \& Fyiz]] \rightarrow \sim Ciz]$	UI 2
5. $Az \& \exists y[(Ey \& Hy) \& Fyiz]$	ACP
6. $\sim Ciz$	MP 4,5
7. Az	simp 5
8. $\sim(Az \& Miz)$	MT 3,6
9. $\sim Az \vee \sim Miz$	DM 8
10 $\sim\sim Az$	Dn 7
11. $\sim Miz$	DS 9,10

Now we’ve hit the target – the consequent of the conditional statement that we were trying to prove. One interesting thing about this proof is that you do not actually need to get rid of the existential quantifier in line 4. The expression ‘$Az \& \exists y[(Ey \& Hy) \& Fyiz]$’ works as the antecedent of the modus ponens inference that yields line 5. Because the whole thing works together as a chunk, you do not need to decompose it. At any rate, how that we have $\sim Miz$ on line 11, we can wrap up the conditional proof, and then use UG to get the conclusion:

1. $\forall x[(Ax \& Mix) \rightarrow Cix]$	
2. $\forall x[[Ax \& \exists y[(Ey \& Hy) \& Fyix]] \rightarrow \sim Cix]$	
3. $(Az \& Miz) \rightarrow Ciz$	UI 1
4. $[Az \& \exists y[(Ey \& Hy) \& Fyiz]] \rightarrow \sim Ciz]$	UI 2
5. $Az \& \exists y[(Ey \& Hy) \& Fyiz]$	ACP
6. $\sim Ciz$	MP 4,5
7. Az	simp 5
8. $\sim(Az \& Miz)$	MT 3,6
9. $\sim Az \vee \sim Miz$	DM 8
10 $\sim\sim Az$	Dn 7
11. $\sim Miz$	DS 9,10
12. $[Az \& \exists y[(Ey \& Hy) \& Fyiz]] \rightarrow \sim Miz$	CP 5-11
13. $\forall x[[Ax \& \exists y[(Ey \& Hy) \& Fyix]] \rightarrow \sim Mix]$	UG 12

Note that in the move from line 12 to line 13, we just took the free variable ‘z’ and replaced it with the bound variable ‘x’.

This translation and proof procedure is indeed a little bit cumbersome. However, it represents a significant achievement. We started out with an interesting philosophical argument about free will. The argument may have seemed or felt intuitively like a good one. But now we’ve shown, using our formalized language Langerese as a tool, that the argument is valid, and that the conclusion is indeed a logical consequence of the premises.

With the introduction of predicates, relations, and quantifiers, we’ve turned Langerese into a formalized language with truly awesome expressive power. However, in formal logic, it seems like there is always a price to pay. Every time we juice up our system of formal logic, we end up losing some advantage that went along with simplicity. Everything involves some calculated trade-off. Recall from our earlier discussion of propositional logic (i.e., Langerese without the predicates and quantifiers) that there is an important difference between truth tables and natural deduction proofs. Setting aside human limitations (§21), truth tables provide an in-principle decision procedure for both tautologousness (or logical truth) and validity (or logical consequence). If an argument is valid, a properly constructed truth table will say so. And if it’s not, the truth table will say that, too. Our natural deduction proof techniques, by contrast, are *one-sided*. If you can construct a proof properly, then that shows – proves! – that an argument is valid. But if you cannot figure out how to construct the proof, there are two potential explanations for that: The problem might lie with *you*, because you just haven’t figured out how to construct the proof yet. Or the problem might lie with *the argument*, because it's invalid. This means that our natural deduction proof techniques, cool though they may be, are not a real decision procedure for tautologousness or validity. In this one respect, natural deduction proof techniques are inferior to truth tables. Now we've seen (§51) that once we bring in quantifiers, we are leaving paradise, the beautiful garden of truth tables where every logical need is taken care of. When arguments involve quantifiers, there's no way to use truth tables to test them for validity. However, our natural deduction proof techniques still give us a valuable set of tools (as the example in this section hopefully demonstrates) even if we no longer have a decision procedure for validity.

§55. Quantifiers and Negation

|Outline|

It turns out that there are also interesting relationships between the quantifiers. But these relationships only come into focus when we start thinking about how the quantifiers interact with negation. To start with a simple example, suppose we want to talk about things being absurd. We can use the predicate '*A*' for '___ is absurd.' Now consider the following:

$\sim\forall xAx$	"Not everything is absurd."

In that case, we just take a universally quantified statement and stick a negation sign out in front. But the following statement says something totally different:

$\forall x\sim Ax$	"For any x, x is not absurd." "Everything is not absurd." "Nothing is absurd."

So it makes a huge difference whether the negation sign sits outside the quantifier, where it operates on the entire statement, or inside the scope of the quantifier. The same goes for the existential quantifier:

$\exists x\sim Ax$	"There is at least one thing that's not absurd." "Something is not absurd."
$\sim\exists xAx$	"It's not the case that there is anything absurd."

Now as you look at these examples, you may already be noticing something interesting about them. There is not really any difference between saying that something is not absurd, and saying that not everything is absurd. By the same token, there's no difference between saying that nothing is absurd, and saying that it's not the case that there is anything absurd. What this means is that we have a couple of very interesting logical equivalence relations:

$\sim\forall xAx$ is logically equivalent to $\exists x\sim Ax$

and

$\forall x\sim Ax$ is logically equivalent to $\sim\exists xAx$

We can take advantage of these logical equivalences by introducing a new replacement rule: **quantifier negation (QN).** This QN rule actually lets us swap out an existential for a universal quantifier, and vice versa, as long as we take care to bump the negation sign from outside to inside, or vice versa.

There's another interesting equivalence rule that we can take advantage of when doing proofs. The following two statements, on the face of it, appear to be equivalent:

> Everything is absurd.
> It's not the case that there is something that's not absurd.

This gives us another handy rule that we can call **quantifier equivalence (QE):**

> $\forall x Ax$ is logically equivalent to $\sim\exists x\sim Ax$

This comes in a second version as well, since the following two statements are also equivalent:

> Something is absurd.
> It's not the case that everything is not absurd.

So this gets us:

> $\exists x Ax$ is logically equivalent to $\sim\forall x\sim Ax$

QN and QE are rules of logical equivalence, exactly like the replacement rules that we encountered back in §43. But there's a difference. In propositional logic, we basically just relied on truth tables to undergird the logical replacement rules: If two statements have the same truth value on every line of a truth table, then they are logically equivalent. With quantifiers, however, we've moved beyond the simple, elegant world of truth tables. If someone asked to you to back up the QN and QE rules, that would take more doing. I've tried to introduce the rules here in an informal way that makes them seem obvious. But if you have been paying any attention at all (for example, in §9), you should know that in logic, obviousness does not count for much! Ultimately, what's needed here is a more rigorous metalogical proof of QN and QE. If it helps, consider that when we were earlier on relying on truth tables, we were basically using them to generate metalogical proofs for our inference and replacement rules. Ideally, we should be able to provide the same thing here, but without relying on the truth tables. This is another moment where I shall kick the can down the road a ways. Rather than going to the trouble to prove QN and QE, I propose that we just rely for the time being on this informal introduction, while bearing in mind that ultimately, more rigor is needed.

QN and QE together enable us to perform a cool trick on longer quantified statements. The trick is called moving in the tildes, or more generally, moving in the negation signs. For the moment, the trick might seem pretty useless. And to be honest, for most purposes you won't need it. But it still reveals something interesting about Langerese. Here's how it works. Suppose you have a longer statement such as the following:

> $\sim\forall x\exists y\sim\exists z\,[(Fx \,\&\, (Gxy \vee Hxz)]$

For present purposes, we can just ignore the stuff on the right that's highlighted. We won't be making any changes to it at all. Focus instead on the quantifiers. If you look at the part highlighted below, it's clearly a negated existential. But we know from QN that '$\sim\exists z$ ...' is equivalent to '$\forall z\sim$ ...'

$$\sim\forall x\exists y\sim\exists z\,[(Fx \,\&\, (Gxy \vee Hxz)]$$

If we apply QN to the highlighted quantifier, we can move the tilde to the right while switching the quantifier from an existential to a universal:

$$\sim\forall x\exists y\forall z\sim[(Fx \,\&\, (Gxy \vee Hxz)]$$

Now check out the quantifier at the left:

$$\sim\forall x\exists y\forall z\sim[(Fx \,\&\, (Gxy \vee Hxz)]$$

We can apply QN there again:

$$\exists x\sim\exists y\forall z\sim[(Fx \,\&\, (Gxy \vee Hxz)]$$

This is a little boring, but if we keep on doing this over and over, we can move that tilde, step by step, all the way to the right!

$$\exists x\sim\exists y\forall z\sim[(Fx \,\&\, (Gxy \vee Hxz)]$$
$$\exists x\forall y\sim\forall z\sim[(Fx \,\&\, (Gxy \vee Hxz)]$$

Here's the next step:

$$\exists x\forall y\sim\forall z\sim[(Fx \,\&\, (Gxy \vee Hxz)]$$
$$\exists x\forall y\exists z\sim\sim[(Fx \,\&\, (Gxy \vee Hxz)]$$

Now we can use double negation to get:

$$\exists x\forall y\exists z\,[(Fx \,\&\, (Gxy \vee Hxz)]$$

This trick of moving in the tildes points to one really intriguing feature of Langerese: Any time you have a quantified statement with tildes mixed up among the quantifiers, that statement is going to be logically equivalent (via repeated applications of QN or QE) to some other statement where all the tildes have been moved in.

§56. Translation Holism

|Outline|

Translation into Langerese with predicates and quantifiers can get pretty complicated. But it can also be fun. There may well be occasions when reasonable people can disagree about which is the best way to translate a statement from English into Langerese. Such disagreement is not necessarily a bad thing at all. It could be that these are precisely the cases where Langerese is quite useful, because it helps to disambiguate statements in the natural language. Anyway, there actually are quite a number of standard, conventional approaches to translation. Once you learn what these are, translation becomes a lot easier. You can handle the straightforward cases very rapidly, and then use your mental energy to try to figure out the trickier ones. The following table gives a bunch of standard approaches to translation.

English statement	Langerese statement(s)	How to read it:
Toby is a good dog.	$Dt \,\&\, Gt$	Toby is good, and Toby is a dog.
All dogs are fuzzy	$\forall x(Dx \rightarrow Fx)$	For any x, if x is a dog, then x is fuzzy.
Some dogs are mean	$\exists x(Dx \,\&\, Mx)$	There is at least one x, such that x is a dog and x is mean.
No dogs are cats	$\sim\exists x(Dx \,\&\, Cx)$ or $\forall x(Dx \rightarrow \sim Cx)$	There is no x, such that x is a dog and x is a cat. Or: For any x, if x is a dog, then x is not a cat.
Some dogs are not playful	$\exists x(Dx \,\&\, \sim Px)$	There is at least one x, such that x is a dog and x is not playful.
Toby sometimes plays with Shiloh	$\exists x(Tx \,\&\, Ptsx)$	There is at least one x, such that x is a time, and Toby plays with Shiloh at x.
Toby only plays with happy dogs.	$\forall x[Ptx \rightarrow (Dx \,\&\, Hx)]$	For any x, if Toby plays with x, then x is happy and x is a dog.
Toby sometimes pesters Shiloh	$\exists x(Tx \,\&\, Ptsx)$	There's some x, such that x is a time, and Toby pesters Shiloh at x.

One question that you might run into in predicate logic is whether to use a predicate *vs.* an individual constant to translate something. In some cases, this is really obvious. For example, when you have a proper name, like 'Toby', just use an individual constant. And when you talk about "some dogs" it makes sense to use "dogs" as a predicate. But other cases can be rather tricky. For example:

"Hope springs eternal."

Here the predicate (in the ordinary grammatical sense) is “springs eternal.” So we can make things a bit easier on ourselves by just using a one-place predicate ‘S’ to mean “springs eternal.” (If we take “springs eternal” to mean something like “springs at all times,” then we could use a universal quantifier; for simplicity’s sake, though, set that aspect of the translation to one side and treat “springs eternal” as a one-place predicate.) The tricky part is figuring out how to translate the subject of this sentence, “hope.” Consider two possibilities:

(1) Sh

(2) $\exists x(Hx \& Sx)$

The first approach here treats hope as an individual thing in our domain of discourse, and just assigns a constant ‘h’ to hope. The second approach treats “____ is hope” as a predicate. There is some x, such that x is hope, and x springs eternal. Which is the better way to handle a case like this may depend a lot on the context. When translating a statement in isolation, it may not matter much how you do it. However, the whole point of formal logic is to shine light on the logical relationships among statements, so when handling multiple statements, you’ll want to translate them in way that helps to clarify those relationships. Consider:

Hope is a delusion.
Toby hopes.
Therefore, Toby is deluded.

This looks like a valid argument. But the most natural translation makes it hard to see the validity:

D_1h
Ht
Therefore, D_2t

The subscripts are needed here because there is a major difference between saying “x is a delusion” and “x is deluded.” But the main problem here is that in the first premise, we’re using an individual constant for ‘hope’, while in the second premise, we’re using a predicate. The two premises do not mesh logically. And as translated, the argument is obviously invalid. But according to the principle of charity (§37), we should strive to find a way of translating the argument that makes it come out valid. A good first step is to translate “hope” in the same way throughout:

$\exists x(Hx \& D_1x)$
Ht
Therefore, D_2x

There is still a problem, though, since we have a predicate in the conclusion – D_2 – that doesn’t show up anywhere in the premises. So the argument still is not valid. Another glitch has to do with the predicate ‘H’. If we take the first premise to say, “Something is both a hope and a delusion," then the second premise has to translate as "Toby is a hope," which gets things wrong. However, it's possible to smooth out these wrinkles in the following way. We can reframe the first premise, so that it has the same predicate as the conclusion. That is, we could take "Hope is a delusion" to mean "Anyone who hopes is deluded." Then we get the following elegant translation:

$\forall x(Hx \rightarrow Dx)$
Ht
Therefore, Dt

Now the validity of the argument comes into sharp focus. It's easy to prove the conclusion using UI and *modus ponens*. Of course, we had to purchase formal validity at a price: our translation of the first premise may sacrifice a bit of accuracy for the sake of making the argument come out valid.

Fascinatingly, there is actually another, completely different way of making the argument come out valid:

Dh	Hope is a delusion.
Hth	Toby has hope.
Therefore, $\exists x(Htx \& Dx)$	Therefore, Toby has something that's a delusion.

This is also valid. In order to prove the conclusion, all you need to do is conjoin the two premises, and then use the EG rule. The translation of the conclusion is a little strained, for sure. But this approach is also justifiable by appeal to the principle of charity, since it makes the argument come out valid. Recall that we started with the question about how to translate "hope." Should we use a predicate or an individual constant? It turns out that either approach works, *provided that you translate everything else in a certain way.*

The above reflections point in the direction of a view that we might call **translation holism**. According to translation holism, the proper "units" of translation are not individual statements, but rather packages or bundles of statements. Translation holism follows from the fact that much of what we care about in formal logic concerns relationships among statements (validity, logical consistency, etc.), together with the principle of charity. As the example above also shows, the correct translation of a set of statements is sometimes underdetermined. That is, there will sometimes be multiple translations that are equally good, in the sense that they preserve the validity of the argument equally well.

Translation holism has another interesting consequence. It's tempting to think of translation of an argument (into Langerese) and proving the conclusion of that argument as two totally distinct steps in a process. But translation holism means that they are not really distinct. In the example above, we actually relied on our understanding of various inference rules (UI, *modus ponens*, conjunction, and EG) in figuring out how to translate the argument.

- Translation holism

Technical Term

§57. Translation Technicalities

|Outline|

In the previous section, the examples involved only one quantifier. Sometimes, though, we might need two quantifiers. For example:

> At least one of the dogs plays with all the dogs.

When faced with a longer statement like this, it can sometimes help a lot to work stepwise toward a good translation. For example, "at least one" definitely calls for an existential quantifier. So we might start with:

> There is at least one x, such that x is a dog and x plays with all the dogs.

Now we can symbolize this:

> $\exists x[Dx$ & x plays with all the dogs]

Call this a partially translated expression. Partially translated expressions are what you get when you take a natural language sentence and use logical symbols to capture just part of the logical structure. Note that these partially symbolized expressions are not wffs at all. If we were constructing proofs, such partially symbolized expressions would be pretty useless for that reason. However, they can be really, really helpful when it comes to translation. They can also be helpful with respect to illustrating the logical structure of claims and arguments. Sometimes, a partial translation can reveal something interesting about that structure, or can help to disambiguate a statement that is a little unclear. In the above case, all we need to do in order to finish the translation above is to come up with a good translation for "x plays with all the dogs." But that is easy:

$$\forall y(Dy \rightarrow Pxy)$$

For any y, if y is a dog, then x plays with y. Of course, the above string of symbols is not really a statement, since 'x' is a free variable. But that's no problem, since the whole idea is to plunk this down inside the partially symbolized expression that we started with:

$$\exists x[Dx \,\&\, \forall y(Dy \rightarrow Pxy)]$$

Now 'x' is bound by the existential quantifier on the left. And we have an elegant translation of the statement, "At least one of the dogs plays with all the dogs."

You might well be wondering if there could be other ways of translating this. For example, what about:

$\exists x \forall y[Dx \,\&\, (Dy \rightarrow Pxy)]$

This one reads a bit differently:

> There is at least one *x*, such that for any *y*, *x* is a dog and if *y* is a dog, then *x* plays with *y*.

The only difference between these two translations is the placement of the universal quantifier. It turns out that the two translations are logically equivalent, which means that either of them works just fine. The first one tracks the grammatical structure of the English sentence a bit better. But if you think things through, you can see why both translations work. The only real difference between them is whether '*Dx*' falls under the scope of the universal quantifier. But since the universal quantifier doesn't apply to '*x*' at all, this difference turns out to make no difference. This case is another example of the general rule that logically equivalent translations are equally good. Also, if you care to, you can *prove* that they are logically equivalent. Here is one half of the proof:

1. $\exists x[Dx \,\&\, \forall y(Dy \rightarrow Pxy)]$	Show: $\exists x \forall y[Dx \,\&\, (Dy \rightarrow Pxy)]$	
2. $Da \,\&\, \forall y(Dy \rightarrow Pay)$	EI 1	
3. $Da \,\&\, (Dz \rightarrow Paz)$	UI 2	
4. $\forall y[Da \,\&\, (Dy \rightarrow Pay)]$	UG 3	
5. $\exists x \forall y[Dx \,\&\, (Dy \rightarrow Pay)]$	EG 4	QED

Notice that in constructing this proof, we adhered to all the restrictions discussed earlier (§§52 and 53). On line 2, we used a new letter, '*a*'. And on line 3, we instantiated using a variable, '*z*' rather than a constant. That made it possible to generalize again on line 4. To prove that these two statements are equivalent, all you need to do is run the proof in the other direction.

The above example shows that it is sometimes allowable to put a quantifier in different places. Now you might also be wondering about the ordering of the quantifiers. Does the ordering matter? The answer is sometimes yes, sometimes no. The easiest way to see why is to consider some examples:

> Somebody barks at somebody.
>
> $\exists x \exists y Bxy$

What happens to this statement if we switch the order of the quantifiers?

$\exists y \exists x Bxy$

Interestingly, this makes no difference at all to the truth-value of the statement. Either way, we are saying, "There is at least one *x*, such that …" and "There is at least one *y*, such that …" The ordering of the quantifiers makes no difference here.

What about the ordering of the variables? At first glance, it might seem like the ordering of the variables must matter a lot. There is a huge difference between the following two statements:

$\exists xBtx$ “Toby barks at somebody”

$\exists xBxt$ “Somebody barks at Toby.”

However, if we are seeking to translate “Somebody barks at somebody,” the ordering of the variables doesn't really matter. Either of the following works fine:

$\exists x\exists yBxy$

$\exists x\exists yByx$

Both of these are equally good ways of saying: There are two “bodies” such that one somebody barks at the other somebody.

Now suppose we wanted to say that everybody barks at everybody:

$\forall x\forall yBxy$

Here again, the ordering of the quantifiers and the variables doesn't really make any difference. These examples point toward a general rule of translation, which is that when you have string of existential quantifiers only, or a string of universal quantifiers only, the ordering of the quantifiers does not actually matter. However, when you have a mix of existential and universal quantifiers, the ordering matters a great deal. Consider the following example:

$\forall x\exists yBxy$

This says: everyone barks at someone. Or spelled out a bit more carefully: for any x, there is some y, such that x barks at y. But if you change the order of the quantifiers while leaving everything else fixed, you get a rather different statement:

$\exists y\forall xBxy$

This says: someone is barked at by everyone. Or again: There is some y, such that for any x, x barks at y. If you think it through, the two statements say very different things:

“Everyone barks at someone”
“Someone is barked at by everyone.”

Suppose that we are quantifying over the dogs at the park: Toby, Shiloh, Skipper, and Gracie. It's easy to come up with a scenario where the first statement is true but the second is false. If Toby barks at Shiloh, while Shiloh barks at Skipper, Skipper barks at

Gracie, and Gracie barks at Toby, then everyone barks at someone. But no one dog is barked at by everyone. This shows that the two statements are not logically equivalent.

- Partially translated expression

Technical Term

§58. Identity

|Outline|

Suppose we want to translate the following statement into Langerese:

> "Toby plays will all the dogs other than Skipper."

It might be tempting to try to say: "For any *x*, if *x* is a dog, then Toby plays with *x*, but Toby does not play with Skipper." One obvious problem, though, is that if Skipper is a dog, then this statement will be self-contradictory.

$$\forall x[(Dx \rightarrow Ptx) \; \& \; {\sim} Pts]$$

This says: "For any *x*, if *x* is a dog, then Toby plays with *x*, but Toby doesn't play with Skipper." But if Skipper is a dog, then this can't be right. We need some way of translating "other than." Now, saying that *x* is "other than" *y* is a way of saying that *x* and *y* are not the same. They are not identical. This suggests one thing that might help here: we could introduce a new relation of identity.

As it happens, since we already have a good handle on relations (§49), introducing one new relation is no big deal. We can just use the letter '*I*' for identity if we want. You may remember that every relation is interpreted by getting assigned a set of ordered pairs. That makes the interpretation of identity really easy. We can just assign to '*I*' the following set of ordered pairs:

<Toby, Toby>
<Skipper, Skipper>
<Gracie, Gracie>
<Shiloh, Shiloh>
<Daisy, Daisy>
Etc.

Everything, in other words, is identical to itself. As easy as this sounds, many philosophers and logicians have thought that identity is not just any old binary relation. Instead, there is something quite special about it.

First, though, it's important to note that "identity" has meanings in ordinary contexts that have nothing to do with its meaning in logic. Often, when people talk about someone's identity, they mean (very roughly) that someone's personality, or that someone's sense of who they are. Identity in the logical sense has absolutely nothing to do with personality. We could say, in the logical sense, that a rock is identical to itself, even though it has no personality.

Second, it's important to appreciate that even in technical logical and philosophical contexts, identity has at least two different senses. We need to distinguish numerical identity from qualitative identity. To say that *x* and *y* are numerically identical is to say that they are one and the same thing. For example: Toby is one and the same thing as Toby. Plato is one and the same thing as the founder of the Academy in Athens. To say that two things are qualitatively identical is to say that they are exactly alike. To give a couple of plausible examples: two pennies from the same mint are numerically distinct (they are not one and the same thing—there are two of them!) but qualitatively identical, because they are exactly alike. A painting and a perfect forgery of that painting are also qualitatively identical but numerically distinct. (Of course, if you think hard about it, you could ask awkward questions about these examples. Perhaps the two pennies are not exactly alike because they have different spatial locations. But hopefully the examples convey the basic idea.) Or consider the case of something persisting over time. Suppose my car gets a dent in the door. The car post-dent is numerically identical with the care pre-dent – it's one and the same car. But the pre-dent and post-dent cars are qualitatively different, thanks to the dent. Just how quantitative and qualitative identity are related to one another is itself a fascinating philosophical issue, and one the pursuit of which could take us pretty far afield. For the time being, just note that in logic, numerical identity is the sort of identity that counts. It's what we were aiming to capture with the predicate '*I*'. From here on out, the term 'identity' will mean 'numerical identity.'

Suppose we wanted to say that Toby is numerically identical with Toby:

$$Itt$$

That comes out true, given our interpretation of the predicate '*I*', because the ordered pair <Toby, Toby> falls under the extension of that binary predicate. But there is still a problem with this way of approaching things. This makes it seem like '*Itt*' is a statement that just happens to be true—a contingent statement, as it were. But this does not seem right. "Toby is (numerically) the same as Toby" seems like a tautology. Indeed, some might even call '*a* is identical to *a*' the "law of identity." Surely we do not need to know anything about an item in order to know that it is one and the same thing as itself. In other words, it seems like identity is a very special sort of binary relation. One common way to handle this is to introduce a new logical operator: identity. Using this new operator, '=', if we want to say that Toby is (numerically) the same as Toby, we can write:

$$t = t$$

Or suppose I have another nickname for Toby: "Toby-wan Kenobi." If we wanted to say that Toby just *is* Kenobi, we could write:

$$t = k$$

If we wanted to say that Toby is not (numerically) the same as his friend Shiloh – they are different dogs – we could write:

$$\sim(t = s)$$

But there is also another customary way of doing this:

$$t \neq s$$

Either way of writing negated identity statements is fine, though the negated identity symbol ('≠') will be convenient. Now recall that any time we introduce a new logical operator, we have to give a semantic theory for it. With identity, that is fairly easy to do, along the following lines:

> '$a = b$' is true under interpretation $\Im$ if and only if $\Im$ assigns the same item from the domain of discourse to 'a' that it assigns to 'b'.

This is perhaps a strange case, where we are using a new logical operator to capture a binary relation, when we already have a way of doing that with relational predicates. However, identity is a very special relation, in part because of the power that it adds to Langerese.

Identity is an example of what is sometimes called an equivalence relation. This is a fancy way of saying that it has three particular features that set it apart from many other binary relations. The first feature is reflexivity. A relation is reflexive when everything stands in that relation to itself. This is just another way of saying that '$a = a$' is a tautology. Everything stands in the relation of numerical identity to itself. The second feature is symmetry: If $a = b$, then $b = a$, and vice versa. Compare this to other binary relations, such as "older than." I cannot be older than myself, so reflexivity fails. Symmetry fails too: It is obviously not the case that if I am older than you, you are also older than me. Finally, identity exhibits transitivity: If $a = b$, and $b = c$, then $a = c$. The "older than" relation is also transitive, but many other binary relations fail transitivity. For example, consider the "can't stand" relation. If *a* can't stand *b*, and *b* can't stand *c*, it does not follow that *a* can't stand c. Overall, then, the specialness of the identity relation—its reflexivity, symmetry, and transitivity—might be a reason to give it pride of place and introduce '=' as a new logical operator. As it happens, identity is going to be the last logical operator that we introduce into Langerese.

Every logical operator requires introduction and elimination rules. These rules enable us to actually use the operator when doing natural deduction proofs. Happily, with identity, the rules of inference are very straightforward. Consider first identity elimination:

$$\begin{array}{ll} a = b & \\ \underline{Fa\qquad\qquad} & \qquad =\text{E} \\ \text{Therefore, } Fb & \end{array}$$

This rule (=E) is closely related to G.W. Leibniz's principle of the indiscernibility of identicals. Here "indiscernibility" means qualitative identity: to say that two things, *a* and *b*, are indiscernible means that whatever is true of *a* must also be true of *b*. Our elimination rule (=E) is based on this idea. If $a = b$, then the thought is that any other statement that's true of *a*, such as *Fa*, must also be true of *b*. (A quick aside: Leibniz's more famous principle is the identity of indiscernibles. That one runs the other way, saying that if two things are such that whatever is true of the one is true of the other—if they are indiscernible—then they are also numerically identical. In order to formulate that principle, though, we need even fancier logical techniques, since we'd have to quantify over predicates.)

How about the introduction rule for identity? We're going to stick with a really easy and boring introduction rule. Since '$a = a$' is a tautology, for any constant we wish, we can just use this as a way of introducing identity *ex nihilo*:

$a = a$ =I

Here it makes no difference what the earlier lines in the proof might be. This is because a tautology follows logically from any set of premises whatever (§42).

What justifies these new rules, =E and =I? I won't spell out the metalogical justification here fully, but both of these rules are rooted in our semantic theory for identity, or the interpretation of identity offered above. For example, if our interpretation $\mathfrak{I}$ assigns the same item from the domain to 'a' that it assigns to 'b', and if 'Fa' is true, then it's easy to see that 'Fb' must also be true.

Identity does not complicate our proof techniques much at all, but it does open up a whole new world of translational possibilities. I will not try to be comprehensive here; however, a couple of examples will show how much additional translational heft we get with identity.

To start with, identity gives is a powerful way of translating statements involving superlatives. Suppose, for example, we want to say that Toby is the fastest dog at the park. Up to now, our toolbox for translating statements like this has been fairly limited. We could use the one-place predicate "_____ is fast." Or we could bring in a two-place predicate "______ is faster than ______". If need be, we could just introduce another one-place predicate " _______ is the fastest." But the problem with that approach is that "fastest" is also clearly a relational notion. "Fastest" means something like "faster than all the others." And this is where identity comes in. Suppose we tried to translate "Toby is the fastest dog" in the following way:

$$Dt \,\&\, \forall x(Dx \rightarrow Ftx)$$

Toby is a dog, and for any x, if x is a dog, then Toby is faster than x. This is on the right track, but there is a problem, since it clearly implies that *Toby is faster than himself*. But that makes no sense. What we want to say is that Toby is faster than all the *other* dogs. And identity enables us to do that.

$$Dt \,\&\, \forall x[(Dx \,\&\, t \neq x) \rightarrow Ftx]$$

This says: Toby is a dog, and for any other x, where x is a dog other than Toby, Toby is faster than x. This strategy can be used more generally to translate any superlatives.

I'll conclude this section – and our exploration of Langerese – with just one more example of the added translational power that we get with identity. Bertrand Russell famously showed how definite descriptions can be translated into formal logic. An *in*definite description is just a descriptive phrase that begins with 'a' or 'an,' such as:

A barking dog . . .
An enthusiastic retriever . . .

It is actually fairly easy to see how to translate these indefinite descriptions using the existential quantifier. For example, the statement

> A barking dog next door is giving me a headache.

Can be rendered as:

$$\exists x[Bx \,\&\, Dx \,\&\, Nx \,\&\, Gxhm]$$

We can read this as: There is at least one thing such that it's a barking dog, and it's next door, and it's giving me a headache. But what if we wanted to translate what is known as a definite description, with a 'the':

> The barking dog next door is giving me a headache.

It is much trickier to see how to do this. Russell's great insight was that the identity relation can help clarify the difference between indefinite and definite descriptions. The definite description ("The barking dog next door …") isn't only saying that there is at least one barking dog next door, although it *is* saying that. It is, in a sense, referring to just one dog in particular—namely, the one that's barking next door. And this referring to one dog in particular is what makes the definite description different from an indefinite one. And crucially, identity gives us a way of zeroing in on one particular thing.

To take this step by step, let's begin by thinking about how to translate a slightly different statement:

> There is only one dog.

Suppose that, tragically, at some point in the future dogs are nearly extinct, and there is only one left. We can translate this in the following way:

$$\exists x[Dx \,\&\, \forall y(Dy \to y{=}x)]$$

This says, roughly: There's at least one thing that's a dog, and if anything else is a dog, then it's identical to that one thing. This is a clever way of expressing the idea that there's at least one dog, and there are no others than that one. This clever move can help with definite descriptions, too.

To return to the main example:

> The barking dog next door is giving me a headache.

To break this down a bit, we can treat it as saying: There is some *x*, such that *x* is a barking dog next door. Moreover, there's no other barking dog next door. And that barking dog next door is giving me a headache. If we use identity in the same clever way that we just did with the other example above, we get the following translation:

$$\exists x[(Dx \,\&\, Bx \,\&\, Nx) \,\&\, \forall y((Dy \,\&\, By \,\&\, Nx) \to y{=}x) \,\&\, Gxhm]$$

I've left out a few parentheses to make things easier to see. Overall, this has a fairly simple structure:

$$\exists x[\ \underline{\hspace{3cm}}\ \&\ \underline{\hspace{3cm}}\ \&\ \underline{\hspace{2.7cm}}\ ]$$

Then we add the first part:

$$\exists x[[(Dx \,\&\, Bx \,\&\, Nx)\ \&\ \underline{\hspace{3cm}}\ \&\ \underline{\hspace{2.5cm}}]$$

This just says that there's at least one dog barking next door. Next, we can add the part about the headache:

$$\exists x[(Dx \,\&\, Bx \,\&\, Nx)\ \&\ \underline{\hspace{3.8cm}}\ \&\ Gxhm]$$

Now we've got: There's at least one thing, such that it's a dog barking next door and it's giving me a headache. This is basically what we'd want if we were translating an indefinite description. But we can make it definite by adding that there's just one dog barking next door:

$$\exists x[(Dx \,\&\, Bx \,\&\, Nx)\ \&\ \forall y((Dy \,\&\, By \,\&\, Nx) \rightarrow y=x)\ \&\ Gxhm]$$

Now, at last, we have: There's at least one x, such that it's a dog barking next door, and there's no other dog barking next door, and it's giving me a headache.

Russell's development of this approach to translating definite descriptions, in his famous 1905 essay, "On Denoting," was of considerable historical importance. It gave new impetus to analytic philosophy, and it helped convince many philosophers that the tools of formal logic could fruitfully be brought to bear on questions in metaphysics and the philosophy of language.

Technical Terms

- Equivalence relation
- Identity of indiscernibles
- Indiscernibility of identicals
- Numerical identity
- Qualitative identity
- Reflexivity
- Symmetry
- Transitivity

§59. Form and Content: The Value of Formal Logic

|Outline|

At this point, our survey of basic formal (classical) logic is complete. We now have at our disposal a well-developed formal language, Langerese. Langerese includes truth-functional logical operators ('~', '∨', '&', '→', and '↔'), as well as individual constants and variables, predicate constants, quantifiers, and identity. The semantic theory for the truth-functional operators is provided by truth tables. The semantic theory for individual constants and predicates was sketched in §48. The semantic theory for the quantifiers was sketched in §51. And finally, the semantic theory for identity ('=') was supplied in §58. Along the way, at various points, we also supplied syntactical or grammatical rules for Langerese, rules that specify what counts as a wff. We did this a bit more carefully for the truth-functional operators (§6). At other points along the way, we just looked at some examples of syntactical rules without fully and carefully spelling them out. But these, too, would be easy to supply. Furthermore, we have at our disposal a full set of inference and replacement rules for constructing natural deduction proofs in Langerese, including introduction and elimination rules for all of the truth-functional operators, for the existential and universal quantifiers, and for identity. This system of formal logic—often called classical logic—serves as a kind of starting point for any and all further investigation in logic.

Because it takes so much effort and practice, along with a fair amount of memorization, to become proficient with a formal system such as Langerese, it is completely reasonable to ask what the point is of going to so much trouble. This is one of those cases where it's really hard to address this "what's the point?" question before you begin. But now that we are at least familiar with Langerese, we can consider some possible answers.

(1) One possible answer is simply that understanding formal validity, or formal logical consequence, is absolutely crucial for evaluating many arguments. In all of our thinking and reasoning together, we make arguments. And formal validity, though it's neither necessary nor sufficient for having a good argument, is nevertheless an important aspect of many good arguments. It's one feature that we might aim at when trying to construct logically good arguments. And natural deduction proofs in Langerese afford a way of establishing formal validity. This alone makes Langerese a powerful and useful tool of reasoning. Any time we wish to construct a formally valid argument, we can translate it into Langerese and try to establish its validity, as in §44. Of course, when constructing arguments, one need not aim for formal validity. Many deductive arguments are only materially valid (§30). And many, many of the arguments that we use in ordinary life are inductive. Even the best inductive arguments fall short of validity. It would be a mistake to conclude, though, that the tools for establishing formal validity are just irrelevant to the assessment of inductive and other informal arguments. What Langerese really gives us is a kind of foil or contrast case, so that when we enter the messier, more complicated world of inductive and informal logic, we can better navigate through the messiness and complexity. In studying patterns of better or worse inductive and informal reasoning, it can be helpful to compare/contrast those with the formal patterns (both the valid argument forms and the invalid ones).

(2) Formal logic also has considerable intrinsic aesthetic value. Susanne Langer, after whom I've named "Langerese," argued that the arts – from painting and drawing, to dance, from sculpture to music – all involve the manipulation of symbolic formal structures.[3] Human beings, Langer held, are symbol-using creatures, and systems of symbols always have formal structure. Learning about an art form – for example, studying music – involves learning about the rules that govern those formal transformations, and studying relationships between form and content. For example, a G chord sounds one way on a piano, and another way when sung by an ensemble. Same form, different content. In learning to play an instrument, one becomes more and more proficient at recognizing and producing formal patterns. What's true of music and other art forms is arguably true of logic: The formal patterns themselves are intrinsically worth studying and appreciating. Exploring, studying, appreciating, and creating formal patterns are self-rewarding activities. Appreciating the formal structure of an elegant proof is perhaps not too far removed from the appreciation of beautiful forms in other artistic and creative domains.

(3) Formal logic is all about abstraction and interpretation (§2). Arguments have both form and content, and one purpose of a formal language like Langerese is to give us a way of stripping away the content so as to look at the forms all by themselves. On the other hand, we've seen repeatedly that the whole purpose of giving a semantic theory for a formal language like Langerese is to show how to interpret it, how to give it meaning and content. One of Langer's great insights is that abstraction and interpretation are the two central activities of human cognition. Formal logic is really just one of many examples of a place where abstraction and interpretation are important. One advantage of studying formal logic is that it is especially easy to see how abstraction and interpretation work in this particular case. In logic, one can get a perfectly clear view of how abstraction and interpretation work. This is partly because Langerese is, when all is said and done, a fairly simple and pristine formal system. Once we see how abstraction and interpretation work here, then that might deepen our understanding of how abstraction and interpretation work in lots of other contexts. Logic is an easy place to study the relationship between symbolic form and content. But that form/content relationship is important in virtually every aspect of human cognitive and creative life.

(4) I hesitate to claim that studying logic makes one better at reasoning – say, better at spotting invalid arguments, and better at making valid ones. If you approach formal logic with the goal of becoming a "more logical person," you may end up either disappointed or self-deluded. You might get really good at doing formal proofs, and might even enjoy them, but once you leave the context of a logic course and return to ordinary life, you might find that translating arguments into Langerese is often time consuming and not too helpful when reasoning on the fly. And the quality of your reasoning on the fly might not be much better than it was before. Without regular practice, it's only natural to forget some of the inference and replacement rules. And much of our reasoning on the fly involves arguments (e.g. inductive arguments) that do not lend themselves very well to formalization anyway. Realistically, the ordinary person is not likely to actually use these formal tools very much. Embarrassingly, even those of us who really love logic are very likely to continue to commit fallacies with some frequency when reasoning about mundane things. If studying logic helps you develop skills of pattern recognition—if you find that you are now able to recognize disjunctive syllogisms in the world around you—that's wonderful. However, I'm not sure

---

[3] See Susanne K. Langer, *Philosophy in a New Key*, 3rd edition (Harvard University Press, 1957).

that individual skill development is the most important thing to focus on. One could also seek to "get good at" logic for the wrong reasons. For example, if one approached the study of logic with the goal of seeming more logical than the opposition—if the focus were still on winning the argument or the debate—then one would not have really appreciated what formal logic is all about.

We live in a world where the social and financial rewards often go to those who traffic in bad arguments. In some areas of human life (advertising and politics, for example), disregard for norms of good reasoning is *de rigeur*. This means that part of the value of formal logic is social and ethical. It might be a mistake to think that mastering the ins and outs of Langerese will make you a more logical person, but it might nevertheless in some sense make you a better person by leading you to look at arguments in a new way. Instead of debate between two sides with opposing views, where each side treats arguments as bludgeons for beating the other into submission, or as ways of demonstrating to onlookers that "we" are smarter than "they" are, studying formal logic gives you a different perspective on arguments. Joint inquiry takes the place of debate. Arguments become interesting objects of study in their own right. We can come together to study arguments just as botanists study plants, or as we might study different artworks in a gallery. And once we stop caring about who "wins" the debate, we can look together at the formal structure of an argument with an eye towards truth. If the premises are true, must the conclusion also be true? Perhaps more than anything else, it's this logical ethos that's really valuable. It's an ethos that is reflected in the principle of charity (§37). And it has everything to do with the contrast between logic and rhetoric. It's an ethos that has as much to do with our relationships with other people as it does with arguments. It's an ethos of conciliation and cooperation that treats arguments as shared foci of study and appreciation. Part of the value of formal logic, then, is that studying it can be a way of cultivating this ethos.

(5) There are some areas of philosophy in which formal logic is sometimes very helpful as a research tool. This consideration is less important than the others, because it's less relevant to those who may not be studying philosophy. Nevertheless, it is worth mentioning. Sometimes, formal logical analysis can help clarify interesting philosophical issues and puzzles. Sometimes, very interesting problems only come into view with the help of formal logic. One example is C.G. Hempel's raven paradox.

Hempel was interested I what philosophers of science call confirmation theory. This, very roughly, is the effort to spell out what it takes for scientific evidence to confirm (or disconfirm) a theory. The raven paradox has to do with the confirmation of scientific generalizations, such as "All *F*s are *G*s," or "All ravens are black," or "All the planets in the solar system have elliptical orbits." We saw (in §50) how to symbolize these generalizations using the universal quantifier:

"All ravens are black."
$\forall x(Rx \rightarrow Bx)$

One very simple, straightforward principle of confirmation theory is that a generalization is confirmed (or gets evidential support from) one of its instances. So for example, if you observe a black raven, that lends a small amount of support to the claim that all ravens are black. Of course, the following argument is not a terribly good one:

<u>There is a black raven.</u>

So, all ravens are black.

It's obviously not valid, and it is not even inductively strong. For a strong inductive argument, we'd need a much bigger sample. The idea, rather, is just that observing the black raven counts as positive evidence for the generalization, even if the degree of evidence if very slight. Holding other things fixed, the overall level of support for the generalization is (slightly!) higher after we observe the black raven than it was before.

So what's the problem? Throughout our explorations of Langerese, we've been treating logically equivalent statements as basically interchangeable. That is the idea behind the logical replacement rules (§43). Thus, for all intents and purposes, there's no difference between 'p & q' and 'q & p.' This would seem to apply to confirmation theory as well. If we have evidence that confirms or supports 'p & q,' then that very evidence must also confirm 'q & p.' After all, evidence is just something that gives us reason to think that a claim is true, but since they are logically equivalent, any reason for thinking that 'p & q' is true will also be a reason for thinking that 'q & p' is true. So far, so good. But a strange problem arises when we think about the statement that all ravens are black. The problem has to do with transposition:

'$p \rightarrow q$' is equivalent to '$\sim q \rightarrow \sim p$'

Transposition means that the following two generalizations are logically equivalent:

$\forall x(Rx \rightarrow Bx)$	"All ravens are black."
$\forall \mathrm{x}(\sim Bx \rightarrow \sim Rx)$	"All non-black things are non-ravens."

It is easy to show, using universal instantiation and generalization along with transposition, that each of these statements logically implies the other. But that means, in turn, that anything that counts as confirming evidence for one should count as confirming evidence for the other. It is easy to find confirming instances of '$\forall x(\sim Bx \rightarrow \sim Rx)$.' Anything that's a non-black, non-raven, would confirm it. So for example, a yellow coffee mug, or a green car, would both lend some small amount of confirming support to the claim that all non-black things are non-ravens. Bizarrely, thanks to transposition plus our assumption that evidence in favor a statement counts equally in favor of anything that's logically equivalent to it, a green mug should also count as confirming evidence for the statement that all ravens are black. At this point, however, it really seems that something has gone badly wrong. Intuitively, it seems crazy to say that a green mug counts as evidence for the generalization that all ravens are black.

What makes the raven paradox paradoxical is that a set of seemingly obvious premises or assumptions leads inexorably to a conclusion that is just as obviously unacceptable. When faced with a puzzle like this one, you have to decide whether to embrace the crazy-sounding outcome, or whether to reject one or more of the assumptions that lead inexorably to it. This is a challenging problem indeed. Rather than taking time here to explore possible responses to it, I just want to make a couple of observations about it. First, the relevant notion of confirmation is not deductive, so this is not a straightforward case of using our proof techniques to show that some argument in science or common sense is valid. Nevertheless, concepts from formal logic are needed in order to bring the problem into focus in the first place. In particular, we need the technical notion of logical equivalence, plus the rule of transposition. The fact that formal logic helps us to see fascinating puzzles like this is a good reason to study it.

(6) One of the best reasons to study Langerese is simply that it gives rise to fascinating new questions, questions that could scarcely have occurred to anyone in the absence of a fully developed formal system. These are the kinds of questions that continue to occupy logicians and philosophers of logic. This might seem like the very opposite of what should motivate us to develop a system of formal logic. Intuitively, the main reason why we should want such a system is to be able to *answer* questions about arguments. The fact that the system *generates lots of new questions* would hardly seem to count in its favor. But this may well be the strongest consideration in favor of studying a system like Langerese. Doing so opens up a whole world of new philosophical questions. This is a point about philosophical fruitfulness.

It's worth considering some of the decision points and forks in the road that we encountered in the development of Langerese. Some of these decisions – such as how many truth values to permit, and whether to avail ourselves of any non-truth functional operators—were quite momentous. At each decision point, we considered the pros and cons of different approaches. For example, allowing three truth values might make it easier to handle vagueness (a pro) but at the cost of making our truth tables far more complicated (a con)—see §4. The general point is that many aspects of Langerese are optional. Nothing prevents us from doing things differently. This fact alone raises profound questions about the status of Langerese. Is it, in some sense, the "best" system of logic? If so, what makes it the best? (And what standards should we apply when determining which system is the best?) Is there a single system of logic (whether Langerese or some other) that tells us what validity *really is*, and which statements really are tautologies? Or might different systems of logic, constructed on different principles, be serviceable for different purposes? Is a system as simple as Langerese sufficient for capturing most of our reasoning in science and common life, or is more complexity needed?

I won't try to answer these questions here. But I do wish to observe just how thin the philosophical air can get when we start exploring them. Philosophers of logic make arguments *about logic*—for example, arguments about the status and features of Langerese and other related systems. In developing such meta-level arguments about logic, one has no choice but to rely implicitly on norms of good reasoning. But of course, part of the point of a system of formal logic is to characterize some of those norms—the ones concerning validity—with as much clarity and rigor as possible. In reasoning about logic, we have no choice but to rely implicitly on assumptions about the difference between good and bad arguments. Thus, where the air is thin, you might experience some philosophical vertigo. But there are also scenic vistas. Circularity looms—How do you make arguments about validity without assuming the very norms that are in question?—and it might seem like there is no firm resting place. But one benefits from expanded conceptions of what reasoning itself might amount to.

(7) Finally, the system of formal logic that I've been calling 'Langerese' is, in a sense, a historical item. Indeed, it is arguably one of the more significant achievements of the human mind, rather like a great scientific theory or a great work of art or musical composition. Like a scientific theory or work of art, it can be appreciated on its own, in isolation from other things—see point (2) above—but it can also be understood in its historical context. In this text, I have not said too much about the history of logic,

mathematics, and philosophy in the late nineteenth and early twentieth centuries, when classical logic was developed, refined, and consolidated. And I have not said too much about how philosophers, such as the logical positivists of the Vienna Circle in the 1920s and 1930s, excitedly put the new formal logic to work in surprising ways. Or about the relationship between formal logic and the development of computing technology in the twentieth century. The thought I wish to share here in closing is just that some very important intellectual developments over the last two centuries are just impossible to understand without first having a good working knowledge of Langerese. One very good reason for studying formal logic is thus historical: large swaths of recent intellectual history are barely intelligible without it.

These, then, are some possible answers to the question about the point or value of studying formal logic. The answers offered here are surely not exhaustive. But if this study of formal logic has helped you to reach the point where you can at least feel the force of some of these considerations, then the main goal of this textbook has been achieved.

Technical Definitions

|Outline|

Abstraction (§2) The process of setting aside or bracketing content (or substance) in order to appreciate somethings formal structure.

Analytic statements (§24) Statements that are true by definition. Tautologies are examples of analytic statements, but they aren't the only ones. Another classic example which is not a tautology is "All bachelors are unmarried men." According to Kant's influential account, an analytic statement is one where the concept of the predicate is already contained in the concept of the subject. But this doesn't cover formal tautologies very well. Tautologies are **formally analytic**, while "All bachelors are unmarried men" is **materially analytic**.

Affirming the consequent (§30). An invalid argument form that, confusingly, looks a lot like modus ponens. It has the form: $p \rightarrow q$; q; therefore, p.

Antecedent (§1) In a conditional statement, the antecedent is the component statement that follows the "If …" The antecedent is a part of the larger conditional statement.

Argument (§1) A set of statements, where one or more (= the premises) are meant to provide justification or support for the other (= the conclusion).

Argument form (§2) What you get when you take an argument and replace all the simple statements with statement variables. It's the formal structure of an argument, without any content.

Argument from Analogy (§35) A variety of inductive argument that involves comparing two items or cases. Most arguments from analogy have the following form: A and B are similar with respect to features F_1 and F_2. A also has feature F_3. So B probably has feature F_3 as well.

Artificial language (§3) Any language that people invented. For example, Pig Latin is an artificial language, as is Esperanto. Computer programming languages are artificial languages. Classical symbolic logic (=Langerese) is also artificial.

Axiomatic system (§38) A set of statements that includes a (usually small) privileged for foundational set of statements – the axioms – plus all and only the other statements that follow logically from the axioms via a set of specified inference rules. Axiomatic approaches to logic are usually contrasted with natural deduction.

Binary relation (§49) A relational predicate that applies to two items. For example, "_____ likes _____" is a binary relation that can be symbolized in Langerese as 'Lxy.'

Bivalence (§4) The principle which says that every statement is either true or false. According to this principle, ‘true’ and ‘false’ are the only truth values. Classical logic is bivalent, but some alternative systems of logic drop the assumption of bivalence.

Bound variable (§50) A variable that is governed by a quantifier. For example, in the formula ‘ $\forall xFxy$,’ ‘x’ is a bound variable, but ‘y’ is not.

Categorical logic (§47) A system of logic, originally developed by Aristotle, that focuses on formal relationships among categorical statements.

Categorical statement (§47) Any statement that says something about how two categories (or groups) are related. For example: “All A’s are B’s” and “Some A’s are B’s” are categorical statements.

Causal dependence (§8) Ordinarily when we say “If p, then q,” we assume that p is a cause and q is an effect. However, classical logic allows for conditional statements to be true even when there is no causal relationship between the antecedent and consequent.

Classical logic (§4) The basic system of formal logic that everyone starts with (= Langerese). Key features of classical logic are, first, that it is bivalent, and second, that all the logical operators are truth functional.

Coherence theory of truth (§5) The theory that what makes a statement true (or false) is whether it belongs to a coherent set of statements.

Complex statement (§1) Any statement that contains at least one other statement as a part. Complex statements are sometimes also called **compound statements** or **molecular statements.**

Conceptual Analysis (§15) An effort to find necessary and sufficient conditions for the correct application of a concept. Or giving a definition in terms of necessary and sufficient conditions.

Conclusion (§1) A statement belonging to an argument that the rest of the argument is meant to justify or support.

Conditional proof (§41) A technique for introducing the ‘→’ into a proof. Conditional proof (CP) is, in effect, the introduction rule for the ‘→’. Begin by assuming the antecedent of the conditional you wish to prove. Then prove the consequent. This shows that if the antecedent is true, then the consequent is true, which means that the conditional statement is true.

Conditional statement (§1) Any statement having an ‘if … then …” structure, sometimes also called a hypothetical statement.

Conjunction (§10). See the truth table for '&'

Conjuncts (§10). Two statements combined to form a conjunction using the '&' operator.

Consequent (§1) In a conditional statement, the consequent is the statement that follows the "then …" The consequent is a part of the larger conditional statement.

Consistency (§28). A set of statements is logically consistent iff it is possible for them all to be true.

Contingent being (§12). Something that might or might not exist. For example, humans might not have existed. So we are all contingent beings.

Contingent statements (§23). Statements that might be true and might be false. A contingent statement will be true on at least one line of a truth table, and false on at least one line. Note that every simple statement is contingent.

Contradiction (§27). Two statements are contradictory iff they never have the same truth value.

Correspondence theory of truth (§5). The theory that what makes a statement true or false is whether it agrees with/corresponds to reality.

Corresponding biconditional (§26). A complex statement that's formed by taking two logically equivalent statements and linking them with a triple bar.

Corresponding conditional (of an argument) (§30). A complex statement that is constructed by taking the premises of an argument, conjoining them, and making them the antecedent of a conditional statement, while making the conclusion the consequent. The corresponding conditional of every valid argument is a tautology.

Decision procedure (§19). Any test or operation that can be used to tell whether something has a property that you are interested in. For example a litmus test is a decision procedure for the presence of acid. Truth tables are a decision procedure for validity.

Deductive argument (§32). Any argument that aims at validity. That is, any argument where the arguer's goal is to support a conclusion by showing that it follows logically from a set of premises.

Deviant logic (§4) An alternative system of logic that departs from classical logic in some significant ways, often by dropping either the principle of bivalence (to get three-valued logic) or the principle of non-contradiction (to get paraconsistent logic).

Dilemma (§40) Any disjunctive either/or statement could be considered a dilemma in the broad sense. However, the term is often used to refer to an either/or where both options have undesirable or unpalatable consequences. The two options are sometimes called the

horns of the dilemma. The inference rule constructive dilemma is a helpful way to represent dilemmas

Disjunction (§13) See the truth table for ‘∨’.

Disjuncts (§13) Two statements combined to form a disjunction using the wedge operator.

Domain of discourse (§48) The set of all the things that you might be talking about in a given context. Or the set of all the things that you are quantifying over. Unless otherwise specified, the domain of discourse can be assumed to be the universal set (or the set of everything).

Elimination rules (§38) Any inference rules that enable you to remove or eliminate a logical operator. For example, *modus ponens* is an elimination rule for ‘→’. Also known as exit rules.

Equivalence relation (§58) Any binary relation that exhibits reflexivity, symmetry, and transitivity. Identity is an equivalence relation.

Exclusive categories (§34) Two categories are exclusive when nothing can fall under both of them. For example: square and triangle are exclusive categories, since nothing can be both.

Exclusive Disjunction (§13) An ‘or’ statement where it is suggested that one thing or another is true but not both.

Exhaustive categories (§34) Two or more categories are exhaustive when everything falls under at least one of them. For example: “living things” and “nonliving things” are exhaustive categories.

Explosion (§31) The principle that any argument with inconsistent premises is valid. This means that a contradiction logically implies anything and everything.

Extension (of a term) (§48) The set of things to which a term refers. For example, the extension of the term ‘dog’ is just the set of all dogs: {Toby, Shiloh, Skipper, Lassie, . . . }

Existential import (§51) A statement has existential import when it explicitly or implicitly asserts the existence of something.

Extensional interpretation of predicates (§48) In predicate logic, it is customary to interpret each monadic predicate constant by assigning to it an extension – a set of items from the domain of discourse to which the predicate applies. A 2-place (binary) predicate gets assigned a set of ordered pairs from the domain, and in general, an *n*-place predicate gets assigned a set of ordered *n*-tuples.

Evidential Strength (§35) An evidentially strong inductive argument is one where it would be improbable for the conclusion to be false while all the premises are true. If the premises are true, then probably the conclusion is true.

Existential quantifier (§50) A logical device ('∃x') that makes it possible to assert the existence of something in the domain of discourse. '∃*xFx*' can be read, loosely, as saying that there is at least one thing in the domain with feature *F*.

Expansion (of a quantifier) (§51) Where the domain of discourse is finite, the expansion of a quantifier is a complex statement, either a conjunction or disjunction, that is equivalent to a quantified statement. For example, if the domain of discourse includes only three items, {*a, b, c*}, then '∀*xFx*' is equivalent to '*Fa* & *Fb* & *Fc*.'

Formalist reading (of a statement) (§17). A way of reading a formula of Langerese where all you do is name the symbols. For example: "Open bracket A arrow B close bracket ampersand tilde A."

Formally analytic statement (§24). A tautology, or any statement that's true in virtue of the definitions of its logical operators.

Formal consistency (§28). Logical consistency in virtue of form. Formal consistency can be assessed using truth tables. A set of statements is formally consistent iff there is some assignment of truth values (some interpretation) to their component statement constants that makes every statement in the set true.

Formal consistency of all simple statements (§28) A feature of classical propositional logic. Any set of simple statements will be formally consistent, because there will be exactly one line on the truth table where they are all true.

Formal contradiction (§27) Contradiction in virtue of form. Two statements are formally contradictory iff there is no line on the truth table where they have the same truth value.

Formal equivalence (§26). Logical equivalence in virtue of form. Two statements are formally equivalent iff there is no lone on the truth table where they have different truth values.

Formal validity (§30) A formally valid argument is one whose validity depends on the logical form or structure of the premises and conclusion. Formal validity can be assessed using truth tables.

Formal validity requirement (§33) A rule saying that you should never make invalid deductive arguments. (The rule allows invalid arguments if they aren't deductive.)

Free Variable (§50) An individual variable that is not bound by a quantifier. For example, in the formula ‘*Gxa*’, ‘*x*’ is a free variable. A formula that contains a free variable is not a genuine statement.

Function (§7) Any operation that takes certain things as inputs and converts them to certain outputs.

Fuzzy logic (§4) An alternative (or deviant) system of logic that drops the principle of bivalence. Instead of just two truth values, fuzzy logic allows for indefinitely many truth values, ranging from 0 to 1. The basic idea of fuzzy logic is that truth and falsity come in degrees.

Generalization rule (§52) An introduction rule for quantifiers (either EG or UG). Generalization rules permit one to infer a quantified statement from one of its instances.

Identity of Indiscernibles (§58) The principle that qualitative identity implies numerical identity. If *a* and *b* are exactly alike (qualitatively identical, or indiscernible) then they are numerically identical.

Inclusive Disjunction (§13) An ‘or’ statement where it is suggested that one thing and/or another might be true. In Langerese, disjunction is treated as inclusive.

Inconsistency (§28) A set of statements is logically inconsistent iff it is not possible for them all to be true.

Indiscernibility of Identicals (§58) The principle that numerical identity implies qualitative identity. If *a* and *b* are one and the same thing, then they are exactly alike (indiscernible).

Individual constant (§48) In predicate logic, an individual constant is any lower-case italicized letter, such as ‘*a*,’ ‘*b*,’ ‘a_1,’ etc., though not including the letters ‘*x*’, ‘*y*’, or ‘*z*’. Individual constants are combined with predicates to form wffs. And each individual constant is interpreted by being assigned one item from the domain of discourse.

Individual variable (§50) In predicate logic, italicized lower-case letters ‘*x*’, ‘*y*’, and ‘*z*’ serve as individual variables. If more are needed, subscripts may be used. Individual variables are uninterpreted placeholders. They do not stand for any particular item(s) in the domain of discourse. They may be either free or bound.

Inductive argument (§32, §35) Any argument that’s not deductive. An inductive argument involves the suggestion that if the premises are true, then the conclusion is probably true.

Inductivism (§35) Inductivists hold that inductive reasoning is generally more important than deductive reasoning in natural science and common life.

Inference rules (§38) Usually contrasted with replacement rules. An inference rule, in the context of natural deduction proof, is a rule that says what conclusion follows logically from a set of premises. Unlike replacement rules, inference rules only go in one direction, and the premises must be entire lines in the proof.

Inference to the best explanation (§35) A common form of inductive reasoning that is sometimes also called "abduction." Arguments to the best explanation typically go as follows: *E* is the best of a range of potential explanations of some phenomenon or dataset. So probably, *E* is true.

Instantiation rule (§52) A rule (UI or EI) that permits one to replace a quantified statement with one of its instances; an elimination rule for quantifiers.

Intension (of a term) (§48) The set of features that speakers commonly associate with a term. Roughly, the meaning of a term. For example, the intension of the term 'dog' might include: furry, barking, quadruped, mammal, tail-wagging, ball-fetching, etc.

Interpretation (of a formula in Langerese) (§2, §7) Interpretation is the opposite of abstraction. To interpret a statement of Langerese, you uniformly assign statements to each of its statement constants. (There is an alternative view of interpretation which says that it involves assigning truth values to each statement constant. But you don't need to worry about this detail.) Interpretation has to do with semantics: you are spelling out the meaning or content of a statement in Langerese when you interpret it.

Introduction rules (§39) Inference rules that permit the introduction of a new logical operator into a proof. For example, addition is an introduction rule for '$\vee$'.

Invalidity (§30). An invalid argument is one where it is possible for all of the premises to be true while the conclusion is false.

Law of excluded middle (§19). This is often confused with the principle of bivalence. The law of excluded middle is a tautology of classical logic: '*A* or not *A*' is an instance of it.

Law of non-contradiction (§23). Another tautology of classical logic: "Not (*A* and not *A*)."

Logical contradiction (§27). Two statements are contradictory when they never have the same truth value.

Logical equivalence (§12). Two statements are logically equivalent when they always have the same truth value. Both are true or both are false.

Logical equivalence of all tautologies (§26) The principle that every tautology is logically equivalent to every other tautology, because they always have the same truth value – True!

Logical operator (§1) A device/symbol that makes it possible to form complex statements out of simple ones, sometimes also called a **logical connective**.

Logically good argument (§37) An argument whose premises actually lend support/justification to its conclusion (as contrasted with a rhetorically good argument).

Main operator (§6). In a complex statement, the main operator is the one that governs the statement as a whole. It is the operator that takes as inputs the largest chunk(s) making up the statement.

Materially analytic statement (§24). A statement that's true in virtue of the meanings of the simple statements that compose it. Any simple statement that's true by definition is materially analytic. An example would be "all triangles have three sides."

Material consistency (§28). A set of statements is materially consistent iff what they say is compatible. For example: "Toby is tired" and "Toby is full of energy" are materially inconsistent.

Material contradiction (§27). Two statements are materially contradictory when they say contradictory things. For example, "Toby is older than Shiloh" and "Shiloh is older than Toby" are materially contradictory, even though there is no formal contradiction between them.

Material Equivalence (§12, §26). This is a term that is sometimes used to characterize the logical operator '↔.' Sometimes material equivalence is contrasted with formal logical equivalence. In this sense, two statements are materially equivalent when their having the same truth value is due to their meaning or semantic content. For example, "I am older than my brother" is materially (but not formally) equivalent to "My brother is younger than me."

Material Implication (§7). See the truth table for '→'

Material validity (§30). An argument is materially valid when its validity depends on what the premises and the conclusion say, or on their content.

Meta-language (§3) The language that you use to say something about another language. For example, if I say "German is easy to learn," I am saying that in English, so English is the meta-language.

Modal operator (§12) A logical operator that modifies a statement in terms of possibility and/or necessity. "Possibly *p*" is a modal operator. Modal logics are systems that add modal operators to classical logic. These modal operators are not, however, truth functional.

Monadic predicate (§49) A one-place predicate, or a predicate that can be applied to just one individual constant at a time. For example, "_____ is a dog" is a monadic predicate.

Natural Deduction (§38) An approach to showing the validity of arguments that makes no claims about the special status of the premises. This approach is usually contrasted with axiomatic systems, and is called "natural" because it (allegedly) better reflects how we actually reason.

Natural language (§3) Any language that people did not invent. So for example, English and Arabic are natural languages. (Note: writing is actually a human invention, but language seems not to be.)

Necessary being (§12) A being that necessarily exists. Or a being that exists in all possible worlds.

Necessary condition (§15) To say that *p* is a necessary condition for *q* is to say that *p* is a prerequisite for *q*. In Langerese, you would write '*q* implies *p*'.

Necessary truth (§12) A statement that's true in all possible worlds.

Negation (§13) See the truth table for '~'.

New constant requirement (§53) When using existential instantiation (EI) in a proof, one must always introduce a new constant that does not yet appear anywhere in the proof.

Numerical identity (as contrasted with qualitative identity) (§58) Two things are numerically identical when they are one and the same thing. For example, Earth is numerically identical with the third planet from the sun: they are one and the same planet.

Object language (§3) The language that you are studying, or the language that you are talking about. For example, if I say "German is difficult," I am using English but talking about German, so German is the object language.

Ordered pair (§49) A pair of items from the domain of discourse, where the order matters. This makes ordered pairs different from sets. The set {Plato, Aristotle} is the same as the set {Aristotle, Plato}. However the ordered pair <Plato, Aristotle> is different from the ordered pair <Aristotle, Plato>. Ordered pairs are useful for giving extensional interpretations of binary relations.

Ordered *n*-tuple (§49) An ordered triple, ordered quadruple, or (more generally) ordered n-tuple, is just like an ordered pair, but with more items. In general, an n-place predicate can be interpreted by assigning to it an ordered *n*-tuple from the domain of discourse.

Paradoxes of material implication (§9) A set of puzzles concerning the truth table for material implication. Basically, this refers to the idea that the conditional, as defined in classical logic, is crazy and makes no intuitive sense. For example, if the antecedent is false, then the entire conditional is true, no matter what the consequent says. So if I said: "If

Harry Potter wrote the Declaration of Independence, then _______,” I could plug in anything at all and the statement as a whole would be true.

Partially translated expression (§57) A statement that is only partially translated from the natural language into Langerese. Partially translated statements are not wffs in Langerese. For example: ‘∃*x*(*x* is a dog and *x* never barks).”

Possible worlds semantics (§12) Because modal operators are not truth-functional, you cannot use truth tables to show how to interpret them. So you need some other sort of semantic theory for modal operators. One standard approach is to talk about logically possible worlds, and to say things like: ‘Possibly *p*’ is true just in case ‘*p*’ is true in at least one logically possible world.

Predicate constant (§48) In predicate logic, a predicate constant is any upper-case italicized letter, such as ‘*F*,’ ‘*G*,’ ‘F_1’, etc. Predicate constants are used (roughly) to translate natural language predicates, such as ‘_____ is a dog’ or ‘_____ is mortal.’ They can be combined with individual constants to form wffs.

Predicate logic (§46) Any system of formal logic that captures the subject-predicate structure of statements. This is usually achieved by using predicate constants and individual constants. Classical logic (Langerese) includes predicate logic.

Premises (§1) The statements in an argument that are meant to do the work of justifying or supporting the conclusion.

Premise-conclusion form (§30) A simple way of writing out arguments, where you number each premise and then use a bar to demarcate the premises from the conclusion.

Principle of Accurate Interpretation (§36) The principle that when reconstructing another’s argument, one should always strive to capture the argument that the person really intended to make, even if the argument is a bad one. Note that this can come into tension with the principle of charity.

Principle of Charity (§36) The principle that when reconstructing another’s argument, one should always cast it in the best possible light. More narrowly, when translating an argument into Langerese, try to resolve any ambiguity or unclarity in a way that makes the argument valid.

Proof (§38) Also called a **derivation.** A proof (in Langerese) is a set of numbered statements, including a set of premises, plus a set of statements, each of which follows from the premises using the inference rules and/or replacement rules. A proof is a way of showing that an argument is valid.

Proposition (§5) The meaning of a sentence or statement, or the thing that the sentence expresses. For example: “I kicked my brother,” and “My brother was kicked by me” are difference sentences, one in the active and one in the passive voice. But they have the same meaning and say the same thing.

Qualitative identity (as contrasted with numerical identity) (§58) Two things are qualitatively identical when they are exactly alike, i.e. when they have exactly the same properties or features.

Quantifier (§50) A logical device that makes it possible to say things about numbers of items. Classical logic uses the universal quantifier (“For any x”) and the existential quantifier (“There is at least one x”). But natural languages include many other quantifiers, such as: some, many, lots, usually, a few, etc.

Recursivity (of grammatical rules) (§6) A syntactical rule lets you start with certain simple wffs as inputs and get a new, complex statement (a wff!) as an output. Recursivity means that you can apply that rule to its own outputs; you can take those outputs and use them again as inputs for making some new wff. More generally, a rule is recursive when it loops back on itself.

Reductio ad absurdum (§45) Sometimes also referred to as indirect proof. This is a special proof technique that involves assuming the opposite of what you wish to prove, and then showing that doing so leads to contradiction. For example, if you wish to prove that P, assume $\sim P$ and show that doing so leads to contradiction. You will thereby have “reduced $\sim P$ to absurdity.”

Reflexivity (§58) A feature of some relations. To say that a relation R is reflexive is to say that everything stands in relation R to itself.

Relation (or relational predicate) (§49) A predicate that applies to more than one item at the same time. For example, “_____ likes _____” is a binary relation that applies to two items. “______ gave ______ to ________” is a three-place relation.

Relevance (§8) One statement is relevant to another if the content or meaning of the one has something to do with the content or meaning of the other. Some logicians don’t like the fact that classical logic allows for valid arguments where the premises are irrelevant to the conclusion. Also, classical logic allows for true conditional statements where the antecedent is irrelevant to the consequent—which is super weird.

Replacement Rules (§43) Usually contrasted with inference rules. In the context of a natural deduction proof, replacement rules allow you to replace any formula (any statement or part of a statement) with another logically equivalent formula. Unlike inference rules, replacement rules are two-way and may be applied to only part of a line in a proof.

Rhetoric (§37) The art of persuasion, or the study of how to make arguments that are good at persuading or convincing people.

Rhetorically good argument (§37) An argument that succeeds in persuading people to accept its conclusion (as contrasted with a logically good argument).

Scope (of a quantifier) (§50) The portion of a formula that a quantifier governs. For example, in the formula '$\forall x(Fx \rightarrow \exists yGyx)$,' the scope of the universal quantifier includes everything within the parentheses.

Self-contradictory statements (§23). Statements that come out false on every line of the truth table. Self-contradictory statements are false under every interpretation, or false no matter how you assign truth values to the statement constants. They are **formal falsehoods**, or false in virtue of their form.

Semantics (§6). The theory of meaning.

Semisound argument (§33). A valid argument whose premises are all reasonable or well-justified, even though one or more of the premises might be false.

Simple statement (§1) A statement that contains no other statements as parts. In old-fashioned works, simple statements are sometimes called **atomic statements**.

Singular statement (§48) A statement that applies a predicate to one particular item. For example: "Socrates is mortal" and "Toby is a dog" are singular statements.

Sound argument (§33). A valid argument with all true premises.

Statement (§4) Any sentence that has a truth value. Declarative sentences are typically statements (though there might be exceptions), but questions and commands are not.

Statement constant (§6). A letter that stands for a particular statement (or truth value). What makes it a constant is that the statement that it stands for never changes. By convention, we'll write these as italicized upper-case letters '*A*,' '*B*,' etc.

Statement form (§4) What you get when you take a statement and replace all the simple statements it contains with statement variables. For example, '$p \vee q$' is a statement form, where '*p*' and '*q*' are variables, not constants.

Statement function (§50) What you get when you take a symbolized statement in predicate logic and replace at least one of the individual constants with an individual variable. For example, '*Dt*' is a statement, while '*Dx*' is a statement function.

Statement variable (§2) A letter that can serve as a placeholder for any statement at all, but which also represents no statement in particular. A statement has no truth value. By convention, we'll write statement variables using lower-case italicized letters, such as '*p*' and '*q*'. Some other logic texts might talk about sentential variables or propositional variables.

Statements only assumption (§52) In the context of natural deduction, this is the assumption that every line in a proof must be a genuine statement. We can relax this assumption when doing proofs with quantifiers, allowing proofs to include lines with statement forms that contain free variables.

Statistical Generalization (§35) A common form of inductive argument that involves reasoning from observed to unobserved cases, usually in the following way: x% of observed A's have feature F. So probably all A's have feature F.

Straw person fallacy (§36) A common informal fallacy that usually fits the following pattern: Someone asserts that P. The arguer then attributes to that person a similar-sounding but less plausible view, P^*. Then the arguer attacks P^*.

Strict implication (§12). A version of implication that's stronger than material implication, because it adds a modal operator. "p strictly implies q" means "Necessarily, p implies q." Some logicians like this because they think strict implication helps avoid the paradoxes of material implication, and that it conforms better to our ordinary intuitions about "if … then …"

Substitution instance (of an argument form) (§2) What you get when you take an argument form and plug in (or uniformly substitute) real statements or statement constants for each statement variable.

Sufficient conditions (§15) To say that p is a sufficient condition for q is to say that p guarantees q. In Langerese, you would write 'p implies q'.

Syllogism (§32) An argument with exactly two premises.

Symmetry (§58) A feature of some binary relations. To say that a relation R is symmetrical is to say that if a bears R to b, then b bears R to a.

Syntax (§6) The theory that specifies the formal structure or grammar of a language. Syntax usually takes the form of a set of grammatical rules.

Synthetic statement (§24) Any statement that's not true by definition. In Kant's classic account, a synthetic statement is one where the concept of the predicate ads something new to the concept of the subject.

Tautology (§19) A statement of Langerese that comes out true on every line of the truth table. A tautology is true under any interpretation, or true no matter how you assign truth values to the statement constants. Tautologies are sometimes called **valid statements**, **theorems of logic**, or **laws of logic**. They are **formal truths**, or true in virtue of their logical form.

Temporal ordering (§8) When we say "If *p*, then *q*" we often assume that *p* has to come before *q* in time. In classical logic, however, a conditional statement can be true even if there is no temporal ordering.

Three-valued logic (§4) An alternative, nonclassical system of logic that allows for three truth values, usually true (T), false (F), and indeterminate (i). This, three-valued logic violates the principle of bivalence.

Transitivity (§58) A feature of some binary relations. To say that a relation *R* is transitive is to say that if *a* bears *R* to *b*, and *b* bears *R* to *c*, then *a* bears *R* to *c*.

Translation holism (§56) The view that the appropriate units of translation (from some natural language into Langerese) are not individual statements but rather arguments or perhaps systems of statements.

Truth bearer (§5). Anything that can have a truth value. This includes statements, sentences, propositions, assertions, beliefs, etc.

Truth function(ality) (§7). A special kind of logical function. It takes the truth values of simple statements as inputs and yields the truth values of complex statements as outputs. The logical operators in classical logic are al truth functions, and a truth table is just a way of showing how truth functions work. To say that a system of logic is *truth functional* is to say that the truth values of the complex statements are uniquely determined by the truth values assigned to all the simple statements.

Truth table (§7). A simple device for representing truth functionality. A basic truth table shows how logical inputs (the truth values of simple statements) generate outputs (the truth values of complex statements. But truth tables are an excellent tool that can be used for other purposes too, like testing arguments for validity.

Truth under an interpretation (§48) A statement is true under an interpretation (of Langerese) if that interpretation assigns truth values to statement letters in a way that makes the whole statement come out true. Or in predicate logic, a statement is true under an interpretation if that interpretation assigns items from the domain of discourse to each individual constant, and assigns extensions to predicate constants, in a way that makes the statement come out true.

Truth values (§4) The most important truth values are 'true' and 'false.' Some logicians think we should allow for more truth values than just these two. But in classical logic, 'true' and 'false' are the only truth values.

UG restriction (§52) Universal generalization (UG) must not be used within a subproof (say, a conditional or reduction proof). It may only be used in a main proof. This restriction is necessary in order to avoid certain invalid inferences.

Universal quantifier (§50) A logical device ('$\forall x$') that enables one to make statements about everything in the domain of discourse. For example, '$\forall xFx$' says that everything as the feature *F*.

Use vs. mention (§3) Most of the time when we say something, we're using terms to talk about other things. For example, if I say "Toby is a dog," I'm using (not mentioning) the term 'Toby', and I'm using it to say something about my dog.

Vagueness (§4) A predicate is vague when its extension is not clearly marked out, or when it is not entirely clear just which things the predicate applies to. The classic example of a vague predicate is baldness: It's easy to see that the predicate applies to some people, and easy to see that it does not apply to others. But there are borderline cases where a person is "sort of" bald, and it isn't clear whether the predicate applies.

Valid argument form (§38) An argument form that is always valid, or valid under any interpretation. Every substitution instance of a valid argument form is valid. One example of a valid argument form is *modus ponens*.

Validity (§30) A valid argument is one such that, if the premises are true, the conclusion must also be true. Alternatively: a valid argument is one where it is impossible for all the premises to be true while the conclusion is false.

Well-formed formula, or wff (§6) Any expression or string of symbols that's grammatically or syntactically correct. For example: "Toby is a dog" is a wff of English—a grammatically correct statement. But "doG Toby, a, ToBy is" is not a wff.