An Introduction to Critical Thinking and Symbolic Logic: Volume 1 Formal Logic

Rebeka Ferreira and Anthony Ferrucci [1]

Preface

This textbook has developed over the last few years of teaching introductory symbolic logic and critical thinking courses. It has been truly a pleasure to have benefited from such great students and colleagues over the years. As we have become increasingly frustrated with the costs of traditional logic textbooks (though many of them deserve high praise for their accuracy and depth), the move to open source has become more and more attractive. We're happy to provide it free of charge for educational use.

With that being said, there are always improvements to be made here and we would be most grateful for constructive feedback and criticism. We have chosen to write this text in LaTex and have adopted certain conventions with symbols. Certainly many important aspects of critical thinking and logic have been omitted here, including historical developments and key logicians, and for that we apologize. Our goal was to create a textbook that could be provided to students free of charge and still contain some of the more important elements of critical thinking and introductory logic.

To that end, an additional benefit of providing this textbook as a Open Education Resource (OER) is that we will be able to provide newer updated versions of this text more frequently, and without any concern about increased charges each time. We are particularly looking forward to expanding our examples, and adding student exercises. We will additionally aim to continually improve the quality and accessibility of our text for students and faculty alike.

We have included a bibliography that includes many admirable textbooks, all of which we have benefited from. The interested reader is encouraged to consult these texts for further study and clarification. These texts have been a great inspiration for us and provide features to students that this concise textbook does not.

We would both like to thank the philosophy students at numerous schools in the Puget Sound region for their patience and helpful suggestions. In particular, we would like to thank our colleagues at Green River College, who have helped us immensely in numerous different ways.

Please feel free to contact us with comments and suggestions. We will strive to correct errors when pointed out, add necessary material, and make other additional and needed changes as they arise. Please check back for the most up to date version.

Rebeka Ferreira and Anthony Ferrucci[1]

[1] To contact the authors, please email: rferreira@greenriver.edu or aferrucci@greenriver.edu. Mailing address: Green River College, SH-1, 12401 SE 320th Street, Auburn, WA 98092

Contents

Chapter 1

Elementary Concepts in Logic and Critical Thinking

1.1 Introducing Logic and Arguments:

Logic, traditionally understood, is centered around the analysis and study of argument forms and patterns. In other words, logic is the study of proper rules of reasoning and their application to arguments. Arguments come in many forms but, as we shall see, we will find it helpful to develop and refine a system of rules and methods that help us deal with language and arguments. More specifically, we want to be able to identify good patterns of reasoning, and crucially, be able to separate them from bad forms of reasoning. This is one of the many ways logic can help. It can give us, in varying degrees of success, methods for improving and evaluating not only our own reasoning, but that of others as well. Given that we are constantly faced and confronted with claims, arguments, and pieces of reasoning, the usefulness of logic can seem all the more appealing.

So if logic is the study of argument forms and patterns, what, then is an argument exactly? An **argument** is a set of sentences, one or more of which we call the premise or premises, which are intended to provide support for or reasons to believe another sentence, the conclusion. In other words, to present an argument is to give reason or reasons for thinking that some conclusion is true. What we do not mean by argument is a quarrel or verbal fight (though arguments can certainly lead to that). Here is an example of an elementary argument:

All states have capitals. Washington is a state. Therefore, Washington has a capital.

As we can see, there are two sentences in the passage here that are intended to provide support for, or reasons to believe, one of the other sentences. The first two sentences give us reason to believe the statement "Washington has a capital." Both the premises and the conclusion are composed of what philosophers call propositions or statements[1]. An argument can have one or more premises but only one conclusion.

By **statement** we mean simply a sentence that is either true or false. To say that a statement is either true or false is just to say that it has a truth value. Bringing this together, an argument then, consists of at least two statements: a statement to be supported (the conclusion) and the statement (premise) meant to support it. Here are some examples of statements:

Olympia is the capital of Washington.
Jupiter is a planet in our solar system.
It has never snowed in San Francisco.
There is carbon based life outside of the solar system.

The first two statements are in fact true while the third statement is actually false. Interestingly, to the fourth statement, while many people may claim that they know the answer, most people understandably will say that they do not know if this statement is true or false. Even in these cases, we still want to say that the statement, on our definition, is either true or false. To see the difference between statements and nonstatements, let's look at the following examples of nonstatements:

[1] There is much philosophical discussion about the distinction between propositions, statements, and claims. While this is an important topic in philosophical logic, we will set aside those discussions here and use the terms interchangeably, but settle by using the word statement throughout. For more of discussion on this topic, see Grayling (2001).

Question: What time is it?
Proposal: Let's go out for lunch.
Suggestion: I would suggest that you practice logic.
Command: Go study!
Exclamation: Awesome!

In these examples, there is nothing claimed by the sentences that is either true or false in the same way as the examples of statements, so we say, for example, that questions do not have a truth value.

Now that we have introduced the idea of arguments being composed of statements, we want to successfully identify the premise(s) and conclusion of an argument. To help us do so often involves indicator words. **Conclusion indicators**, are words used (in our case, English) that lead us to believe that what follows is the argument's conclusion. Here are some of the most commonly used conclusion indicator words:

thus
therefore
accordingly
consequently
hence
it follows that
we may conclude that
as a result

Conversely, we can often identify premises by using indicator words. Common **premise indicators** are:

since
because
given that
for
as indicated by
due to the fact that
this is implied by

To see how these words can help us identify respective premise(s) and the conclusion, let's look at the following argument:

Because Sophie was born in August, it follows that she is a Leo.

Notice that we have two indicator words that tip us off to what the premise and conclusion are. It is worth noting, though, that indicator words may not always be present. Sometimes, we need to assess the relationship between statements in order to determine if an argument is present [i.e., if some statement(s) is meant to support another].

One practice that helps us focus in on an argument's content is called putting an argument into **standard form**. This is just the process of taking an argument in passage form and numbering the premise(s) and conclusion to make argument as a whole clearer to the reader. Let's say that we have an argument contained in the following passage:

> Public schools deserve increased financial assistance. The amount of money spent per student has been decreasing for years in this state. At current funding levels, the state cannot fulfill its constitutional obligation to provide public education to all of its citizens.

Although there are no indicator words in the above example, upon analysis, the second and third statements seem to provide reason to believe the first. By putting this argument into standard form we can better appreciate the flow of the argument:

P1. The amount of money spent per student has been decreasing for years in this state.
P2. At current funding levels, the state cannot fulfill its constitutional obligation to provide public education to all of its citizens.

C. Public schools deserve increased financial assistance.

Notice importantly that the first sentence in our original passage was the conclusion itself, which should always be listed last in standard form. It should be pointed out that while the conclusion may appear at the end of a passage, it can often appear as the first sentence, as in our case above. Additionally, many simple arguments, like the example above, can be combined to form more complex arguments, in which the conclusion of the simpler argument becomes a premise of the larger, more complex, argument.

1.2 Identifying Arguments

Having introduced what an argument is, as well as its constituent parts, it will serve us well to look at some of the differences between arguments and nonarguments. Identifying arguments may not always be easy, but being able to do so, along with discerning which parts provide support for the conclusion, is an important task in logical reasoning. Arguments can be simple or complex, they can be clearly stated or muddled. Even more problematic, is that nonarguments can often times appear quite similar to arguments, and thus be used in their place, rather than having to really provide support for a claim.

What To Look For

Be sure to ascertain if something is being supported. An argument, again at its most basic level, must provide reason or reasons for thinking that some statement is true. If the passage is not doing this, then there is no argument present. To that end, it might be helpful to look at examples of nonarguments to make our concept of argument even clearer.

Nonarguments

Notice that the following, though they may appear to give us reason to think *that* something is the case, in fact merely exclaim, or tell us *how* or *why* something is the case. The following are all examples of nonarguments:[2]

Warnings express danger or alert us to pay special attention, not necessarily providing reason to think that such danger exists.

Advice express recommendations for belief or behavior, without necessarily providing reason to accept the advice.

Beliefs express attitudes towards propositions of acceptance, rejection, or neutrality; without necessarily providing reason to agree.

Opinions express attitudes of judgment or preference, without necessarily providing reason to agree.

Statements though constitutive of arguments, may simply be loosely joined by a similar subject matter, without necessarily providing support for any other statement.

Reports similarly to loosely joined statements, may merely express information about something.

Expositions express greater detail about some statement or subject, without necessarily providing reason to establish the thing being elaborated on.

Illustrations express instances or examples of some statement or subject, without necessarily providing reason to establish the thing being illustrated.

Explanations express descriptions of some event or phenomena, without necessarily providing reason to establish that the thing being explained has occurred.

Conditionals express statements where one part (the consequent) depends upon another part (the antecedent) to hold. These are typically constructed as "if..., then..." statements. They lay out the conditions under which some event or phenomena would hold, without establishing that either conditions actually hold.

Disjunctives express statements where two or more potential options are provided. These are typically constructed as "either..., or..." statements. They lay out the options available, without establishing if either or both actually hold.

It is worth noting that all of the above instances, though insufficient in and of themselves to be arguments, can all be utilized within arguments should they be accompanied by support.

[2] The following list is similar to Hurley (2018), pp. 17-22.

1.3 Deductive and Inductive Arguments

Now that we have seen and discussed the topic of arguments it would help us to make a key distinction between argument types. Traditionally, philosophers and logicians have identified two types of arguments: deductive and inductive. A **deductive argument** is an argument that intends to make the conclusion follow *necessarily* from the premise(s). Here is an example:

P1. All musicians are entertainers.
P2. Regina Spekter is a musician.
C. Therefore, Regina Spekter is an entertainer.

In deductive arguments the intention is that if the reasons (or "premises") are true, then the conclusion must be true. The truth of the premises is meant to establish or guarantee the truth of the conclusion. To determine if the argument is deductive, we can ask ourselves: do the premises attempt to prove the truth of the conclusion?

By contrast, an **inductive argument** is an argument that intends to make the conclusion *likely* or *probable* given the premise(s). The distinction between "necessary" and "likely or probable" conclusions is meant to capture the difference between deductive and inductive arguments. the intention of inductive arguments is that if the reasons (or "premises") are true, the conclusion is only probably true. The truth of the premises is not meant to establish or guarantee the truth of the conclusion, but only make it more likely. Here is an example:

P1. Most musicians are entertainers.
P2. Regina Spekter is a musician.
C. Therefore, Regina Spekter is probably an entertainer.

Here, we can see that even if the premises are true, the conclusion could still potentially be false. To determine if the argument is inductive, we can ask ourselves: do the premises attempt to increase the likelihood of the truth of the conclusion?

Notice the difference between the first argument and the second. In the first case, the conclusion is intended to follow with necessity from the premises. That is, if we accept that "All musicians are entertainers" and that "Regina Spekter is a musician" then what follows is that "Regina Spekter is an entertainer." However, the use of the word 'most' in the second argument leaves open the possibility of the conclusion being false. Both kinds of arguments are used not just in philosophy, but in fields like mathematics, science, and law, just to name a few. It is imperative to understand how each type of argument attempts to guarantee the truth of its conclusion in order to best assess the strength of the arguments one is making or considering.

1.4 Evaluating Deductive and Inductive Arguments

You may have noticed that the aforementioned definitions for deductive and inductive arguments state that they each "attempt" to establish the truth or likelihood of their conclusions, respectively. Here we must note that not all arguments are successful. One of the most significant skills to take away from studying logic and becoming a critical thinker is being able to identify good and bad arguments. Bad reasoning does not establish what it attempts to, and thus, we should not be convinced by it. This does not necessarily mean that the conclusion is false, but that better reasoning is required for it to be established.

Evaluating Deductive Arguments

Valid Arguments

A deductive argument that succeeds in proving its conclusion is said to be **valid**. In a valid deductive argument, it is impossible for true premises to lead to a false conclusion. In other words, the structure of the argument guarantees the truth of the conclusion. To be clear, validity is not grounded in an argument's verified "truth" in the world. Validity merely refers to necessity of the conclusion's truth *if* the premises are true. Again:

P1. All men are mortal.
P2. Socrates is a man.
C. Therefore, Socrates is mortal.

This argument is **valid**. It is impossible for the premises to be true and the conclusion false. In other words, if it is true that "all men are mortal" and that "Socrates is a man", then it must be true that "Socrates is mortal". However, this says nothing about the actual truth of these premises (perhaps the "Socrates" being referred to is my cat).

Invalid Arguments

When a deductively valid argument fails to succeed in guaranteeing the truth of its conclusion, it is said to be **invalid**. In an invalid deductive argument, it is possible for the premises to be true, and the conclusion false. Now let us examine the following argument:

P1. All logic instructors are smart.
P2. Mary is smart.
C. Therefore, Mary is a logic instructor.

Although the structure is similar to the valid argument in the previous example, here, the conclusion certainly does not follow from the premises. That is, it is possible for the conclusion to be false, even if the premises are true.

Key Points: Deductive Validity

1. Validity only applies to deductive arguments, not inductive arguments.
2. Validity refers to the form or structure of the argument, not its content.

Take the following example:

P1. All men have five arms.
P2. Anthony is a man.
C. Therefore, Anthony has five arms.

This argument is valid, even if premise 1 is obviously false. So in order to assess the content of deductive arguments, we need another criterion.

Sound Arguments

Once we have determined that we are dealing with a deductively valid argument, we need to determine whether or not it is sound. An argument is **sound** if and only if it is the case that it is valid and all of the premises are actually true in the world.

Sound Argument = Deductively Valid + All True Premises

Both criteria are important since obviously, truth alone is not enough. It is a mistake to say that an argument is a good one simply because its premises and conclusion are true. Consider this argument:

P1. San Francisco is a city in California.
P2. Seattle is north of San Francisco.
C. Therefore, it rains in Seattle.

Even though every statement in the above argument is true, we could not say that this is a good argument. In order to be sound, not only must the premises all be true, but the conclusion must follow from the premises.

Evaluating Inductive Arguments

Strong Arguments

As discussed in the previous section, unlike deductive arguments, an inductive argument cannot guarantee that if its premises are true, the conclusion will also be true. Even though inductive arguments are not truth-preserving, this does not mean that they cannot still succeed in providing sufficient support for their conclusion. An inductive argument that succeeds in making its conclusion more likely to be true than false is said to be **strong**. Here is an example of a strong inductive argument:

P1. Most *Star Wars* fans dislike Jar Jar Binks.
P2. Rebeka is a *Star Wars* fan.
C. Therefore, Rebeka dislikes Jar Jar Binks.

Common indicator words for strong inductive arguments are:

most
often
almost all

Weak Arguments

Likewise, if an inductive argument fails to make its conclusion more likely to be true than false, it is said to be **weak**. Here is an example of a weak inductive argument:

P1. Some movie theaters are showing *Get Out* every evening this week.
P2. There is a movie theater down the street from my house.

C. Therefore, the movie theater down the street from my house is showing *Get Out* tonight.

Common indicator words for weak inductive arguments are:

a couple
few
some

Key Points: Inductive Strength

1. Strength and weakness only apply to inductive arguments, not deductive arguments.
2. Strength can be subjective, but no matter how strong an inductive argument is, it can never guarantee the truth of its conclusion.

Take the following example:

P1. You are undergoing medical procedure X.
P2. Medical procedure X has a 75% success rate.

C. Therefore, you will have a successful medical procedure.

Although 75% is a rather high likelihood of success, in cases where the stakes are high (say, life and death), some might require a higher bar for strength. However, given the nature of inductive reasoning, even if the success rate was 99.999%, there is no guarantee that the next instance will be successful. For lower stakes content, some may find any likelihood over 50% to be sufficient.

Cogent Arguments

As we saw with validity for deductively valid argument, strong inductive arguments with true premises are said to be **cogent**.

Cogent Argument = Inductively Strong + All True Premises

Even though inductive arguments do not guarantee the truth of their conclusions, even when cogent, it is of great import to establish the likelihood of their success. Induction happens to be the primary means by which we come to know the workings of the empirical world, and is thus one of the bases of scientific reasoning.

Some Problems with Induction

Two primary points of concern for philosophers and logicians with respect to inductive reasoning revolve around its use in explanation and prediction. Obviously, there are many problems with justifying a conclusion based solely on probability. Are we ever justified in moving from a finite number of past observations to predictions about all future observations? How many past observations do we need to make before using them as evidence for universal scientific claims? As critical thinkers, we want strong, well-supported arguments, without making hasty generalizations. Thus, we always want to be careful when arguing about groups as a whole based on a small sample.

Inductive reasoning is also deployed when attempting to determine the most likely explanation for a given phenomena. A common method used for this in science and criminal justice is **Inference to the Best Explanation [IBE]**, where the most likely explanation is asserted as the actual explanation. However, what if our determination of the "best" explanation was selected from wholly bad explanations to begin with? Surely it being the best is by no means any indication that it is true. Just because one has devised an explanation for something does not mean it's the right one. Other explanations, perhaps yet to be considered, could be just as good.[3] As with general problems with induction, IBE always goes "beyond the evidence". It tries to explain facts, but does so by positing a theory that is not derived entirely from those facts.

[3] van Fraassen, p. 143

Guide for Identifying Arguments

Step 1: Are you dealing with an argument?	Arguments					
Step 2: If so, what kind of argument?	Deductive			Inductive		
Step 3: Is the argument successful?	Valid		Invalid	Weak	Strong	
Step 4: Are the premises true?	Sound ✔	Unsound ✗	✗	✗	Uncogent ✗	Cogent ✔

1.5 Deductive Argument Forms

For the remainder of this volume, we will be focusing on the construction and assessment of deductive arguments. Many deductive arguments have a recognizable form, which can aid us in assessing the argument's validity more quickly. These forms refer to their structure, which can be compared to other similar structures of validity. Here are some of the most common deductively valid argument forms that we encounter when studying both formal and informal logic.

Valid Argument Forms

Here is an example of ***modus ponens*** (MP) or affirming the antecedent:

P1. If it is raining, then I need an umbrella.
P2. It is raining.
C. Therefore, I need an umbrella.

which has the following form:

If P, then Q.
P.
Therefore, Q.

Here is an example of ***modus tollens*** (MT), or denying the consequent:

P1. If it is raining, then I need an umbrella.
P2. I do not need an umbrella.
C. Therefore, it is not raining.

which has the following form:

If P, then Q.
Not Q.
Therefore, not P.

Although MP and MT are the most commonly used deductively valid argument forms, they have some dangerously similar invalid counterparts, covered at the end of this section.

Here is an example of ***disjunctive syllogism*** (DS):

P1. We can either have dinner Friday, or Saturday.
P2. We cannot have dinner Friday.
C. Therefore, we can have dinner Saturday.

which can have either of the following forms:

Either P, or Q.
Not P.
Therefore, Q.

or

Either P, or Q.
Not Q.
Therefore, P.

Here is an example of ***hypothetical syllogism*** (HS):

P1. If it is raining, then I need an umbrella.
P2. If I need an umbrella, then I can't carry all of my things.

C. Therefore, if it is raining, then I can't carry all of my things.

which has the following form:

If P, then Q.
If Q, then R.

Therefore, if P, then R.

Here is an example of ***constructive dilemma*** (CD):

P1. If we get a cat, then there will be furballs, and if we get a dog, then there will be fleas.
P2. Either we get a cat, or a dog.

C. Therefore, we will either have furballs or fleas.

which has the following form:

If P, then R, and if Q, then S.
Either P, or Q.

Therefore, either R, or S.

Here is an example of ***destructive dilemma*** (DD):

P1. If we get a cat, then there will be furballs, and if we get a dog, then there will be fleas.
P2. We will have neither furballs, nor fleas.

C. Therefore, we will get neither a cat, nor a dog.

which has the following form:

If P, then R, and if Q, then S.
Neither R, nor S.

Therefore, neither P, nor Q.

Invalid Argument Forms

Here is an example of ***denying the antecedent*** (DA):

P1. If it is raining, then I need an umbrella.
P2. It is not raining.

C. Therefore, I do not need an umbrella.

which has the following form:

If P, then Q.
Not P.

Therefore, not Q.

Notice here that when the antecedent condition (i.e., "it is raining") is not met, nothing can be derived from the conditional statement in the first premise. Who is to say whether or not I need my umbrella when it is not raining? There is not sufficient reason to arrive at the conclusion.

Here is an example of ***affirming the consequent*** (AC):

P1. If it is raining, then I need an umbrella.
P2. I need an umbrella.

C. Therefore, it is raining.

which has the following form:

If P, then Q.
Q.

Therefore, P.

Notice here that even if the consequent condition (i.e., "needing an umbrella") is met, this says nothing about whether or not the antecedent condition (i.e., "it is raining") has also been met. One could possibly need an umbrella even if it were not raining. Again, leaving the conclusion unestablished.

Chapter 2

Propositional Logic

2.1 Introduction to Logical Operators and Translation

By way of introduction, let's say that **propositional logic** is the logic that evaluates propositions or statements[1]. That is, propositional logic gives us the tools to evaluate, compare, and assess the truth value of statements, and arguments composed of statements. More precisely, propositional logic deals with whole or fundamental statements in a way that allows us to formally assess the properties of that language. Propositional logic, then, is our first foray into formal symbolic logic.

Symbols

As was just explained, the focus of propositional logic starts with statements or propositions. In order to best assess the logical structure of propositions, it becomes necessary to translate them from ordinary language into an artificial "language"[2] which will allow us to avoid any confusion or distraction by the argument's content. We will now examine this language's semantics:

Logical Operator	Name	Logical Function	Used to translate
$\sim$	tilde	negation	not, not the case, it is false
&	ampersand	conjunction	and, both
$\vee$	wedge	disjunction	either...or, unless
$\longrightarrow$	arrow	conditional	if...then, provided that, on condition that
$\longleftrightarrow$	double-arrow	equivalence	if and only if, necessary and sufficient

The above semantics will allow us to reproduce even the most complex statements in order to analyze the structure of arguments, and determine their validity.

Translation for Propositional Logic

In order to apply the logical operators above, we first need to understand their role in relation to simple vs. compound statements. A **simple statement** has one subject and one predicate. Each simple statement should be symbolized with a single letter variable (P, Q, R, S, etc.), and if appropriate, that variable should reflect the subject of the statement. If however, there are multiple propositions about the same subject, it is best to use a variable that reflects the predicate of the statement to avoid false equivalency. For example:

Statement	Variable
It is raining	R
I need an umbrella	U
Anthony has five arms	F
Anthony is a philosopher	P

[1]The reader will notice that many systems also refer to propositional logic as *sentential logic*, or the logic of sentences.

[2]As was pointed out in the preface, we have settled on the following symbols merely by convention. Other books and systems use a different set of symbols. We trust that the symbols we have adopted will serve adequately.

Notice that when dealing with simple statements like those above, no logical operators are needed. Logical operators (sometimes also refereed to as 'connectives') are necessary when only dealing with compound statements. A **compound statement** combines at least one logical operator with one or more simple statements. The negation (˜) operator can be attached to a single variable, however, all other operators should be used to connect two variables together. For example:

Statement	**Variables**	**Translation**
It is not raining.	R	∼R
I need an umbrella and a jacket.	U, J	U & J
Either Anthony has five arms, or two arms.	F, T	F∨T
If Anthony is a philosopher, then he cares about logic.	P, L	P⟶L
Anthony will go to the movies, if an only if Rebeka goes.	A, R	A⟷R

Amazingly, all propositions can be translated into this symbolic language. The tricky part is that the same proposition can be written in many different ways. Below are some helpful hints for translating commonly used phrases.

Helpful Hints for Translation in Propositional Logic

Conditionals

In conditional statements (⟶), there are two parts: the **antecedent (if)** and the **consequent (then)**. However, each part may not always be indicated by 'if' and 'then'. Sometimes other words are used, or they may be omitted altogether. **Antecedent indicators** include:

if
provided
whenever

Consequent indicators include:

then
only if

Conjunctions

When conjunctive statements (&) include a negation (∼), it can be difficult to determine whether the negation applies to a single variable, or the entire proposition. For example:

Phrasing	**Translation**
Not both P and Q.	∼(P & Q)
It is both not P and not Q.	∼P & ∼Q

Disjunctions

Similarly for disjunctive statements (∨) and negations. For example:

Phrasing	**Translation**
Not either P or Q.	∼(P∨Q)
Neither P, nor Q.	∼(P∨Q)
Either not P or not Q	∼P∨∼Q
P unless Q	P∨Q

Translating with Multiple Operators

Some statements might be so complex that they have multiple operators, in which case we will need additional punctuation, inducing parentheses ( ) and brackets [ ], to separate the main operator from the secondary operator(s). If you have more than two simple statements, you will be in need of more than one operator. A helpful trick for identifying the main operator is that is will most likely be near the punctuation which bestows the most accurate meaning on the proposition. The main operator will always go outside of the parentheses and brackets. It is also worth restating that the negation (~) operator can be attached to a single variable or on the outside of a statement in parentheses or brackets, while all other operators should only be used to connect two variables together. Finally, parentheses should be used first, with brackets used after. Compare the following examples:

If it is raining, then Anthony will go the movies and Rebeka will stay home.

Variables	**Translation**
R, A, H	R⟶(A & H)

If it rains tomorrow then Anthony will go the movies, and Rebeka will stay home if she has work to do.

Variables	**Translation**
R, A, H, W	(R⟶A) & (W⟶H)

If it rains tomorrow then Anthony will go the movies, and Rebeka will stay home if she has work to do and doesn't feel like seeing a movie.

Variables	**Translation**
R, A, H, W, F	(R⟶A) & [(W & ∼F)⟶H]

Translation in Propositional Logic: Steps 1-4

1. Determine whether or not you are dealing with a simple or compound statement. This will determine whether or not you will be using any logical operators.

Consider the following example: If it rains tomorrow, then Anthony and Rebeka will stay home if they have work to do and there are no good movies playing.

This is a compound statement (and quite a complex one at that).

2. For each simple statement, select the most appropriate variable that best captures the meaning of that statement (either by reference to the statement's subject or predicate). A helpful trick is to list the variables in the order that they appear with space in between to add their connectives, as illustrated below:

Variables:		R		H
Statement:	If	it rains tomorrow,	then	Anthony and Rebeka will stay home
	if	they have work to do	and	there are no good movies playing.
Variables:		W		M

When laid out for translation into symbolic form:

Variables	**Translation**			
R, H, W, M	R	H	W	M

3. For each compound statement, determine which logical operator will be used to connect each constituent simple statement. Be sure to also capture any negations in the statement.

Logical Operators:			⟶	
Statement:	If	it rains tomorrow,	then	Anthony and Rebeka will stay home
	if	they have work to do	and	there are no good movies playing.
Variables:	⟶		&	∼

When inserted into the translation into symbolic form:

Translation						
R	⟶	H	⟶	W	&	∼M

Notice here, that two conditionals, a conjunction, and a negation have been added to the statement. However, we have a problem with our second conditional statement: the last two variables of the statement ('W' and 'M') actually come after the second "if", so they should actually be placed as the antecedent of the second conditional operator while the 'H' variable should be the consequent, as illustrated below:

Translation					
R	⟶	**W**	**&**	**∼M**	⟶ **H**

Here the variables have been rearranged to correctly reflect the semantic meaning of the sentence. This is important since, as we saw earlier in our "Helpful Tips", conditional statements sometimes have the consequent listed prior to the antecedent.

4. For complex compound statements, identify the main logical operator and place parentheses (and if needed, brackets) around each constituent statement. Be sure to identify the main logical operator and ensure that it is placed outside all parentheses and brackets, with the most secondary operators placed within the parentheses. Recall that the main logical operator will most often be placed near the semantically meaningful punctuation of the statement. Remember to begin separating the main logical operator from the secondary operators with parentheses, and then move on to brackets.

Main Logical Operator:		⟶	
Statement:	If it rains tomorrow,	then	Anthony and Rebeka will stay home
	if they have work to do	and	there are no good movies playing.

Notice here that we have begun by identifying the first conditional as the main logical operator since it appears closest to the punctuation in the statement (the comma). This means that the variable 'R' is the antecedent, and the consequent appears to be the entirety of the remaining statement. However, we now know that the three remaining variables on the right side of the main logical operator cannot all be in parentheses (since parentheses can connect two variables at most), so they will be placed in brackets, as illustrated below:

Translation					
R	⟶	[W	&	∼M	⟶ H]

As was just reiterated, we also know that what remains in the brackets will need to be broken up further, since operators can at most connect two variables. This means that we will need to add parentheses around the most secondary compound statement inside the brackets. This change in notation is illustrated below:

Variables	Translation					
R, H, W, M	R	⟶	[(W	&	∼M)	⟶ H]

Notice here that since the variables 'W' and 'not M' form the antecedent of this secondary conditional statement, they will be placed together within the parentheses. As the consequent of this secondary conditional statement, the variable 'H' will go outside of the parentheses, but remain inside of the brackets.

2.2 Truth Functions

After translating statements into propositional logic, a number of assessment methods can be used to verify the potential truth values of those statements as well as the validity of arguments. As discussed in Chapter One, validity concerns the truth-preserving structure of a deductive argument *if* the premises are true. Given the often complex nature of arguments and their constituent parts, it can be difficult to determine just from the statements themselves whether or not they are true, let alone truth-preserving throughout. For instance, if we take the last example from the previous section, it quickly becomes apparent that the truth of the statement depends on a variety of factors:

> If it rains tomorrow, then Anthony and Rebeka will stay home if they have work to do and there are no good movies playing.

Even the truth of more simplistic conditional statements can be difficult to determine when we do not yet know whether any of the conditions hold. It becomes more challenging when there are so many conditions that need to be met, and even more so when we combine a compound statement like the one above with many others in an argument. Luckily, the formal structure of propositional logic allows us to assess the potential truth values of even the most complex statements and arguments. The next section will explore the basic rules for determining the potential truth values of statements, as well as the use of those truth values to determine the validity of arguments.

By **truth value**, we mean the attribution of truth (T) or falsity (F) to a given statement.[3] Simple statements are assigned each possible truth value. The truth value of compound statements will then be determined by the truth values of the simpler statements that make them up. Truth tables are then constructed to determine the truth values of each statement and argument validity. A **truth table** is a representation of the ways that a statement can express truth values. In this sense, truth tables are mechanical and follow rules for their construction.

2.3 Truth Tables for Statements

This section will examine the rules for constructing truth tables for simple and compound statements, as well as instructions for how to classify and compare statements.

Constructing Truth Tables: Steps 1-4

1. Determine the number of simple statements present in the proposition. Remember that each simple statement will be represented by a variable. Each variable will get its own column in the table. For example:

1 variable:

P

2 variables:

P	Q

3 variables:

P	Q	R

2. Determine the number of truth values to be assigned. Remember that each simple statement will be assigned each possible truth value (T and F). Each truth value will get its own row. For compound statements with multiple variables, an easy trick for determining how many rows will be in the table is to multiply 2 (the number of possible truth values) to the power of the number of variables. For example:

1 variable: $2^1 = 2$ rows **2 variables:** $2^2 = 4$ rows **3 variables:** $2^3 = 8$ rows

3. Assign the truth values. In order to make sure all possible arrangements are accounted for in the table, begin on the far left column and divide the rows in half (assigning T to the first half of the rows, and assigning F the second half). Then move to the next column to the right and divide in half again, rotating equally between T and F assignments. For example:

1 variable:

P
T
F

2 variables:

P	Q
T	T
T	F
F	T
F	F

3 variables:

P	Q	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

If you are determining the truth value of a simple statement (one variable), this will be all you need. If you are determining the truth value of compound statements and/or the validity of arguments, you will need to continue.

4. Determine truth values for compound statements. It is worth noting that the rules for each of the following tables will correspond to the logical operator(s) present in each compound statement.

Negation Truth Table: Notice that the truth values for P have been negated. All "not-P" truth values are just the **opposite** of what they would have been for P.

P	$\sim$P
T	F
F	T

Conjunction Truth Table: Notice that the conjunction of "P and Q" is only **true** if both components are true. It is false if either or both components are false.

[3]This text will use the classical notion of truth value, with only two possible values. It is worth noting that in other logical systems (e.g., quantum theory and computer science), there may be more than two possible truth values which allow for different models of entailment, identity, and infinity.

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

Disjunction Truth Table: Notice that the disjunction of "either P or Q" is only **false** if both components are false. It is true if either or both components are true.

P	Q	P∨Q
T	T	T
T	F	T
F	T	T
F	F	**F**

Conditional Truth Table: Notice that the conditional statement of "if P, then Q" is only **false** when the antecedent is true and the consequent is false. It is true in all other instances, even when the antecedent condition does not hold.

P	Q	P⟶Q
T	T	T
T	F	**F**
F	T	T
F	F	T

Equivalence Truth Table: Notice that the material equivalence of "P if and only if Q" is only **true** when both components have the same truth value. It is false if the truth values are different.

P	Q	P⟷Q
T	T	**T**
T	F	F
F	T	F
F	F	**T**

Classifying Statements

Constructing a truth table for compound statements not only helps us to determine the possible truth values of that statement, but can also tell us something about the statement's logical structure. What is more, a truth table for compound statements reveals whether the truth of the statement depends on the specific truth values of its components or the logical form or structure of the entire statement. We can classify even the most complex compound statements in the following ways:

Tautology: All True

Example:

P	Q	[(P⟶Q)	&	P]	⟶	Q
T	T	T	T	T	T	T
T	F	F	F	T	T	F
F	T	T	F	F	T	T
F	F	T	F	F	T	F

Self-contradictory: All False

Example:

P	Q	(P∨Q)	⟷	(∼P	&	∼Q)
T	T	T	F	F	F	F
T	F	T	F	F	F	T
F	T	T	F	T	F	F
F	F	F	F	T	T	T

Contingent: At least One True + At least One False

Example:

P	Q	R	P	⟶	(Q	&	∼R)
T	T	T	T	F	T	F	F
T	T	F	T	T	T	T	T
T	F	T	T	F	F	F	F
T	F	F	T	F	F	F	T
F	T	T	F	T	T	F	F
F	T	F	F	T	T	T	T
F	F	T	F	T	F	F	F
F	F	F	F	T	F	F	T

Notice in the examples above that the determination of classification depends upon the truth values under the main logical operator.

Comparing Statements

Similarly, truth tables may also be used to determine the relationship between multiple statements. This is valuable when considering the strength of an argument and how well various parts of the argument work to support one another. We can compare statements to one another in the following ways:

Logical Equivalence: Same truth value on every line

Example:

P	Q	P⟶Q
T	T	T
T	F	F
F	T	T
F	F	T

P	Q	∼Q	⟶	∼P
T	T	F	T	F
T	F	T	F	F
F	T	F	T	T
F	F	T	T	T

Contradiction: Opposite truth value on every line

Example:

P	Q	P⟶Q
T	T	T
T	F	F
F	T	T
F	F	T

P	Q	P	&	∼Q
T	T	T	F	F
T	F	T	T	T
F	T	F	F	F
F	F	F	F	T

Consistency: At least one line which are both True

Example:

P	Q	P∨Q
T	T	T
T	F	T
F	T	T
F	F	F

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

Inconsistency: No lines which are both true

Example:

P	Q	P⟷Q
T	T	T
T	F	F
F	T	F
F	F	T

P	Q	P	&	∼Q
T	T	T	F	F
T	F	T	T	T
F	T	F	F	F
F	F	F	F	T

Notice in the examples above that the comparison of statements depends upon the truth values under the main logical operators of each statement.

2.4 Truth Tables for Arguments

Having introduced the general method for constructing truth tables and the rules for determining the truth values of compound statements, we can apply the above steps to construct truth tables for arguments. After a table is constructed, we can assess the validity of the argument utilizing the truth table. This is particularly helpful when dealing with deductive arguments that do not conform to the common argument forms introduced in Chapter One.

For this section, we will use the following argument:

If you want to be an astronaut, you must have a background in science or engineering. Unfortunately, you have neither a background in science nor engineering. So you won't be an astronaut.

Translation:
P1. A⟶(S∨E)
P2. ∼(S∨E)
C. ∼A

Constructing Truth Tables: Steps 5-7

5. Determine the number of columns needed for the truth table. This will combine the variables themselves, as well as a column for each premise and the conclusion.

Truth Table: 3 variables: $2^3 = 8$ rows

3 variables + 3 lines in the argument = 6 columns

A	S	E	A ⟶ (S∨E)	∼ (S∨E)	∼A
T	T	T			
T	T	F			
T	F	T			
T	F	F			
F	T	T			
F	T	F			
F	F	T			
F	F	F			

6. Assign truth values to each line in the argument using the rules in step 4. Begin by determining the truth values of the **secondary operators and simple statements first**.

A	S	E	A	⟶	(S∨E)	∼	(S∨E)	∼A
T	T	T	T		T		T	
T	T	F	T		T		T	
T	F	T	T		T		T	
T	F	F	T		F		F	
F	T	T	F		T		T	
F	T	F	F		T		T	
F	F	T	F		T		T	
F	F	F	F		F		F	

Once the truth values for all secondary operators and simple statements have been determined, you will use those truth values to determine the values of the **main logical operators**.

A	S	E	A	⟶	(S∨E)	∼	(S∨E)	∼A
T	T	T	T	T	T	F	T	F
T	T	F	T	T	T	F	T	F
T	F	T	T	T	T	F	T	F
T	F	F	T	F	F	T	F	F
F	T	T	F	T	T	F	T	T
F	T	F	F	T	T	F	T	T
F	F	T	F	T	T	F	T	T
F	F	F	F	T	F	T	F	T

7. Determine the arguments validity. Recall that an argument is only invalid when all of the premises are true and the conclusion is false. Using the truth values of the main logical operators in each premise and conclusion, look for any **lines with all true premises and a false conclusion**. If there are *one or more* such lines, then the argument is **invalid**. If there are *no* such lines, then the argument is **valid**.

A	S	E	A	⟶	(S∨E)	∼	(S∨E)	∼A	
T	T	T	T	T	T	F	T	F	
T	T	F	T	T	T	F	T	F	
T	F	T	T	T	T	F	T	F	
T	F	F	T	F	F	T	F	F	
F	T	T	F	T	T	F	T	T	
F	T	F	F	T	T	F	T	T	
F	F	T	F	T	T	F	T	T	
F	F	F	F	**T**	F	**T**	F	**T**	←

In the example above, we can see that there is only one line with all true premises, and on this line the conclusion is also true. Since there are no lines with all true premises and a false conclusion, this argument is **valid**.

Additionally, consider this argument, similar to, but importantly different from the one utilized above:

If you want to be an astronaut, you must have a background in science or engineering. You have a background in science, but not in engineering. So you will be an astronaut.

Translation:
P1. A⟶(S∨E)
P2. S &∼E
C. A

Applying the steps we have just covered, we can construct the following truth table:

Truth Table: 3 variables: $2^3 = 8$ rows

3 variables + 3 lines in the argument = 6 columns

Beginning with the secondary operators and simple statements, we can assign the following truth values:

A	S	E	A	⟶	(S∨E)	S	&	∼E	A
T	T	T	T		T	T		F	T
T	T	F	T		T	T		T	T
T	F	T	T		T	F		F	T
T	F	F	T		F	F		T	T
F	T	T	F		T	T		F	F
F	T	F	F		T	T		T	F
F	F	T	F		T	F		F	F
F	F	F	F		F	F		T	F

Moving on to the main logical operators, we can assign the remaining truth values:

A	S	E	A	⟶	(S∨E)	S	&	∼E	A
T	T	T	T	T	T	T	F	F	T
T	T	F	T	T	T	T	T	T	T
T	F	T	T	T	T	F	F	F	T
T	F	F	T	F	F	F	F	T	T
F	T	T	F	T	T	T	F	F	F
F	T	F	F	T	T	T	T	T	F
F	F	T	F	T	T	F	F	F	F
F	F	F	F	T	F	F	F	T	F

Having completed the truth table, we can now assess the argument for validity by looking for any lines with all true premises and a false conclusion:

A	S	E	A	⟶	(S∨E)	S	&	∼E	A	
T	T	T	T	T	T	T	F	F	T	
T	T	F	T	T	T	T	T	T	T	
T	F	T	T	T	T	F	F	F	T	
T	F	F	T	F	F	F	F	T	T	
F	T	T	F	T	T	T	F	F	F	
F	T	F	F	**T**	T	T	**T**	T	**F**	←
F	F	T	F	T	T	F	F	F	F	
F	F	F	F	T	F	F	F	T	F	

In the example above, we can see that there is only one line with all true premises, and on this line the conclusion is false. This is sufficient to prove that this argument is **invalid** since it is possible for the premises to be true and the conclusion to be false, even if it is only one line out of eight.

2.5 Indirect Truth Tables

As you may have noticed in the section above, truth tables can be rather arduous, especially when dealing with three or more variables. What is more, many of the lines end up being useless in determining an argument's validity. For longer or more complex arguments with more variables, an alternative method may be preferable to determine validity. In order to more quickly assess an argument's validity, one could opt to use the **indirect truth table** method, where one begins by assuming that an argument is invalid (by assigning all true truth values to the premises and a false truth value to the conclusion) and then works backwards to see if it is in fact possible that those truth values be produced by the logical operators present in each part of the argument. Consider the following argument:

∼P⟶(Q∨R)
∼P
———————
R⟶P

Constructing Indirect Truth Tables: Steps 1-6

1. Begin by aligning all premises and the conclusion horizontally (similarly to long truth tables, but without additional columns for each variable). Separate each premise with a / and the premises from the conclusion with //, as illustrated below:

∼	P	⟶	(Q	∨	R)	/	∼	Q	//	R	⟶	P

2. Assign truth values to the main logical operators in each premise and conclusion that assume that the argument is invalid (all true premises and a false conclusion), as illustrated below:

∼	P	⟶	(Q	∨	R)	/	∼	Q	//	R	⟶	P
		T					T				F	

3. Assign truth values to the variables that would result in the assigned invalid truth values above. The most important part of this process will be to determine if there is a single possible truth value for each variable that will result in the invalid truth values assigned. To determine this, it is best to begin with the simpler statements. This process is illustrated below:

∼	P	⟶	(Q	∨	R)	/	∼	Q	//	R	⟶	P
		T					T	F		T	F	F

Here, the truth values for all three variables have been derived.

4. Apply the truth values for all variables have been determined, to all other presentations of those variables throughout the argument, as illustrated below:

∼	P	⟶	(Q	∨	R)	/	∼	Q	//	R	⟶	P
	F	T	F		T		T	F		T	F	F

Here, all variables have been assigned truth values.

5. Assign truth values to the remaining logical operators of each premise and conclusion that would result in the assigned truth values determined thus far. This should only be done after the truth values for all variables have been determined and assigned. This is a bit different from the long truth table method where you work from the most secondary operator(s) out to the main operator. Here, the process is reversed, beginning from the main logical operator and working in to the most secondary operator(s), as illustrated below:

∼	P	⟶	(Q	∨	R)	/	∼	Q	//	R	⟶	P
T	F	T	F	T	T		T	F		T	F	F

Here, the truth values of all remaining logical operators can be determined using the truth table rules presented in Section Three of this Chapter.

6. Determine the validity of the argument using the indirect truth table method. If the assumed invalid truth values do not lead to a contradiction [where the same variable(s) are assigned the same truth values throughout, and there are no inconsistencies in the truth values of the logical operators], then the argument is **invalid**.

∼	P	⟶	(Q	∨	R)	/	∼	Q	//	R	⟶	P
T	F	T	F	T	T		T	F		T	F	F

∼	P	⟶	(Q	∨	R)	/	∼	Q	//	R	⟶	P
T	F	T	F	T	T		T	F		T	F	F

∼	P	⟶	(Q	∨	R)	/	∼	Q	//	R	⟶	P
T	F	T	F	T	T		T	F		T	F	F

Here, we can see that the truth values for each variable are entirely consistent throughout the table. This shows that the argument is *invalid*, since there is a way for all of the premises to be true and the conclusion to be false without resulting in a contradiction.

If, however, the assumed invalid truth values lead to a contradiction [where the same variable(s) are assigned opposing truth values, or there are inconsistencies in the truth values of the logical operators], then the argument is in fact **valid**.

Consequently, consider this argument:

P⟶(Q∨R)
Q⟶S
P
———
∼R⟶S

Applying the steps we have just covered, we can begin by assigning truth values to the main logical operators that assume that the argument is invalid:

P	⟶	(Q	∨	R)	/	Q	⟶	S	/	P	//	∼	R	⟶	S
	T						T			T				F	

We can then begin determining the truth values of variables which would produce the assumed invalid truth values above:

P	⟶	(Q	∨	R)	/	Q	⟶	S	/	P	//	∼	R	⟶	S
	T						T			T		T	F	F	F

Notice that for this example, the only variables that can be determined at this point are P, R, and S. This is because P presents as a simple statement in premise three and the rules for truth tables provide us with many possibilities of when a conditional is true, but only one for when it is false (when the antecedent is true and the consequent is false). Thus, we do not yet know what truth values to assign to the conditional statements in the first two premises.

Now, taking the determined truth values above, we can apply them to other occurrences of those variables throughout the argument:

P	$\longrightarrow$	(Q	$\vee$	R)	/	Q	$\longrightarrow$	S	/	P	//	$\sim$	R	$\longrightarrow$	S
T	T			F			T	F		T		T	F	F	F

We can now use the truth table rules for the remaining logical operators to determine the remaining truth values, and again apply them to other occurrences of any remaining variables:

P	$\longrightarrow$	(Q	$\vee$	R)	/	Q	$\longrightarrow$	S	/	P	//	$\sim$	R	$\longrightarrow$	S
T	T	F	F	F		F	T	F		T		T	F	F	F

Finally, in assessing the validity of this argument, we can see that, even though the truth values are consistent throughout, there is a logical inconsistency in premise one:

P	$\longrightarrow$	(Q	$\vee$	R)
T	**T**	F	F	F

The antecedent of this conditional statement is true, the consequent is false, yet the truth value under the main logical operator is true (when it should be false). This shows that the argument is *valid*, since there is no way for all of the premises to be true and the conclusion to be false without resulting in a logical inconsistency.

Complex Indirect Truth Tables

Now in some cases, assigning truth values to the argument which assume it is invalid allow for numerous possibilities for what the other truth values could be to produce such invalidity. This most often occurs with conditional and disjunctive statements (which each have three possibilities for being true) and conjunctive statements (which have three possibilities for being false). This means that it may not be possible to determine the single possible truth value for each variable right away (as shown in step 3). In such instances, we will need to account for all such possibilities, which will result in slightly longer indirect truth tables. However, these complex indirect truth tables end up being significantly shorter than the direct truth table method, still making them preferable. Consider the example below:

$\sim$	P	$\longrightarrow$	Q	/	Q	$\longrightarrow$	P	/	P	$\longrightarrow$	$\sim$	Q	//	P	&	$\sim$	Q
		T				T				T					F		
		T				T				T					F		
		T				T				T					F		

Since we know that conditional statement have three possibilities for being true, and conjunctive statements have three possibilities for being false, we have accounted for each above with their own row.

Now we can determine the various possible truth values for the simpler constituent parts of the conclusion, since we know the three instances in which conjunctive statements are false:

$\sim$	P	$\longrightarrow$	Q	/	Q	$\longrightarrow$	P	/	P	$\longrightarrow$	$\sim$	Q	//	P	&	$\sim$	Q
		T				T				T				**T**	F	**F**	T
		T				T				T				**F**	F	**T**	F
		T				T				T				**F**	F	**F**	T

Note, that one could have started with a different part of the argument (e.g., one of the premises), however since there are so many conditional statements in this example, a strategic choice was made to narrow down all possible arrangements of truth values by beginning with conjunctive statement in the conclusion.

Since we know the truth values for all of the variables, we can now extend those to all other presentations of those variables throughout the argument:

$\sim$	P	$\longrightarrow$	Q	/	Q	$\longrightarrow$	P	/	P	$\longrightarrow$	$\sim$	Q	//	P	&	$\sim$	Q
F	T	T	T		T	T	T		T	T	F	T		T	F	F	T
T	F	T	F		F	T	F		F	T	T	F		F	F	T	F
T	F	T	T		T	T	F		F	T	F	T		F	F	F	T

Notice here that each row is assigned the determined truth value of the variables on that line.

Now that all truth values have been assigned, we can assess the argument for validity. This means looking for instances where the assumed invalid truth values lead to a contradiction, either because the same variable(s) have been assigned opposing truth values, or because there are inconsistencies in the truth values of the logical operators.

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

Here, we can see that there are three instances of logical inconsistency, one line under each premise. In each instance, the antecedent of the conditional statement is true, the consequent is false, yet the truth value under the main logical operator is true (when it should be false). This shows that the argument is *valid*, since there is no way for all of the premises to be true and the conclusion to be false without resulting in a inconsistency (in this case, many inconsistencies).

Chapter 3

Natural Deduction for Propositional Logic

We saw in the previous chapter how to establish the validity or invalidity of an argument by using truth tables. In most respects truth tables are mechanical and relatively straightforward. However, truth tables can be cumbersome, as we also saw in the previous examples. Having established the validity of certain argument patterns, we can now use them to show how certain conclusions follow from a given set of premises. Put another way, we now want to use what we can call **rules of inference**, which are discrete logical inferences that allow us to infer some statement from another statement, or set of statements. A rule of inference then, is a "valid move" from one line to the next. In this sense, **natural deduction** is the application of the rules of inference as individual steps to show how a given conclusion follows in a valid argument form. In this regard, natural deduction resembles a "proof" that might be given in geometry, a process that is more illuminating and revealing than what we get by simply doing a truth table. This chapter will introduce the rules of inference allowed in classical logic, as well as how those rules are used to prove a desired propositional statement through natural deduction.

3.1 Rules of Implication

Similar to the deductively valid argument forms covered in Chapter One, the first five rules of implication reflect legitimate methods of inferring a conclusion from certain given premises. These, as well as other **rules of implication**, allow us to imply particular conclusions from a set of given premises. When used in natural deduction, the application of these rules help us to derive a conclusion from premises which may not obviously imply that conclusion. These rules can also be used to derive intermediate premises which, often times combined with other rules, lead to the desired conclusion. It is worth noting that the rules of implication only apply to the main logical operators in each participating premise (and not to the constituent parts of a premise). The rules of implication take the symbolic forms below:

1. Modus Ponens (MP): When dealing with material implications, the consequent can be derived when the antecedent is present. Notice that this rule is only applicable when the antecedent condition is affirmed. It would be an *invalid* step to derive the antecedent condition when the consequent is affirmed.

$$\begin{array}{l} P \longrightarrow Q \\ P \\ \hline Q \end{array}$$

Here are some additional examples of modus ponens in more complex forms:

$$\begin{array}{l} \sim P \longrightarrow (Q \longleftrightarrow R) \\ \sim P \\ \hline Q \longleftrightarrow R \end{array}$$

$$\begin{array}{l} (P \vee Q) \longrightarrow \sim(R \;\&\; S) \\ P \vee Q \\ \hline \sim(R \;\&\; S) \end{array}$$

$$\begin{array}{l} P \,\&\, Q \\ (P \,\&\, Q) \longrightarrow [(R \longrightarrow S) \,\&\, (T \longrightarrow U)] \\ \hline (R \longrightarrow S) \,\&\, (T \longrightarrow U) \end{array}$$

2. Modus Tollens (MT): When dealing with material implications, the negation of the antecedent can be derived when the negation of the consequent is present. Notice that this rule is only applicable when the consequent condition is denied. It would be an *invalid* step to derive the negation of the consequent condition when the antecedent is denied.

$$\begin{array}{l} P \longrightarrow Q \\ \sim Q \\ \hline \sim P \end{array}$$

Here are some additional examples of modus tollens in more complex forms:

$$\begin{array}{l} (P \vee Q) \longrightarrow R \\ \sim R \\ \hline \sim (P \vee Q) \end{array}$$

$$\begin{array}{l} \sim P \longrightarrow \sim (Q \vee R) \\ \sim\sim (Q \vee R) \\ \hline \sim\sim P \end{array}$$

$$\begin{array}{l} \sim P \\ [(Q \vee R) \,\&\, (S \vee T)] \longrightarrow P \\ \hline \sim [(Q \vee R) \,\&\, (S \vee T)] \end{array}$$

3. Hypothetical Syllogism (HS): When dealing with a series of conditional statements, the antecedent of one can imply the consequent of another if they share a variable. Notice that the shared variable must be the consequent of one conditional and the antecedent of another. This rule closely reflects the mathematical law of transitivity (If A=B, and B=C, then A=C).

$$\begin{array}{l} P \longrightarrow Q \\ Q \longrightarrow R \\ \hline P \longrightarrow R \end{array}$$

Here are some additional examples of hypothetical syllogism in more complex forms:

$$\begin{array}{l} P \longrightarrow (Q \,\&\, R) \\ (Q \,\&\, R) \longrightarrow \sim S \\ \hline P \longrightarrow \sim S \end{array}$$

$$\begin{array}{l} \sim P \longrightarrow (Q \longrightarrow R) \\ (S \vee T) \longrightarrow \sim P \\ \hline (S \vee T) \longrightarrow (Q \longrightarrow R) \end{array}$$

$$\begin{array}{l} (P \longrightarrow Q) \longrightarrow [(R \vee S) \,\&\, T] \\ (U \longleftrightarrow V) \longrightarrow (P \longrightarrow Q) \\ \hline (U \longleftrightarrow V) \longrightarrow [(R \vee S) \,\&\, T] \end{array}$$

4. Disjunctive Syllogism (DS): When dealing with a disjunctive statement, one can derive either constituent part as long as the other part is no longer an option. Notice that this rule is only applicable if one of the disjuncts is negated. It would be an *invalid* step to derive one option when the other has been affirmed.

$$\begin{array}{l} P \vee Q \\ \sim Q \\ \hline P \end{array}$$

or

P∨Q
~P

Q

Here are some additional examples of disjunctive syllogism in more complex forms:

P∨~(Q & R)
~P

~(Q & R)

~(P∨Q)
(P∨Q)∨(R⟶S)

R⟶S

~P∨[(Q⟶R) & (S⟶T)]
~~P

(Q⟶R) & (S⟶T)

5. Constructive Dilemma (CD): When dealing with two conditional statements conjoined together, the consequents of each conditional statement can be derived when the antecedents are present. Notice that the antecedents and consequents are presented as disjunctive statements.

(P⟶Q) & (R⟶S)
P ∨ R

Q ∨ S

Here are some additional examples of constructive dilemma in more complex forms:

~P ∨ Q
(~P⟶S) & (Q⟶~T)

S ∨ ~T

[(P⟶Q)⟶(R & S)] & [(T⟶U)⟶(R & V)]
(P⟶Q) ∨ (T⟶U)

(R & S) ∨ (R & V)

6. Simplification (Simp): When dealing with any conjunction, either or both of its constituent parts can be separated out to stand on its own.[1]

P & Q

P

or

P & Q

Q

Here are some additional examples of simplification in more complex forms:

~P & (Q⟷R)

~P

(P∨Q) & (R⟶S)

R⟶S

[(P⟶Q) & R] & (S⟶T)

(P⟶Q) & R

7. Conjunction (Conj): Any two premises can be joined together as long as they are connected by a logical conjunct.

[1] Some textbooks place limitations on the use of simplification and disjunctive syllogism by requiring one to obtain only the left side of the expression when emplyoing such rules. We have ommitted such a requirement here for simplicity (see Hurley).

$$\begin{array}{l} P \\ Q \\ \hline P \mathbin{\&} Q \end{array}$$

Here are some additional examples of conjunction in more complex forms:

$$\begin{array}{l} \sim P \\ \sim Q \\ \hline \sim P \mathbin{\&} \sim Q \end{array}$$

$$\begin{array}{l} P \longrightarrow Q \\ R \longrightarrow S \\ \hline (P \longrightarrow Q) \mathbin{\&} (R \longrightarrow S) \end{array}$$

$$\begin{array}{l} P \longrightarrow (Q \mathbin{\&} R) \\ S \longrightarrow (Q \mathbin{\&} T) \\ \hline [P \longrightarrow (Q \mathbin{\&} R)] \mathbin{\&} [S \longrightarrow (Q \mathbin{\&} T)] \end{array}$$

Notice that when complex statements are conjoined, the conjunct should be placed outside of the parentheses or brackets which separate the two original statements.

8. **Addition (Add):** Any variable can be added to an existing premise as long as it is connected by a logical disjunct.

$$\begin{array}{l} P \\ \hline P \vee Q \end{array}$$

Here are some additional examples of addition in more complex forms:

$$\begin{array}{l} P \\ \hline P \vee P \end{array}$$

$$\begin{array}{l} Q \\ \hline Q \vee \sim R \end{array}$$

$$\begin{array}{l} P \mathbin{\&} Q \\ \hline (P \mathbin{\&} Q) \vee (R \mathbin{\&} \sim S) \end{array}$$

$$\begin{array}{l} R \longleftrightarrow S \\ \hline (R \longleftrightarrow S) \vee [P \longrightarrow (Q \longrightarrow T)] \end{array}$$

3.2 Rules of Replacement

Along with the rules of implication, introduced above, there are additional tools which can be used to derive a conclusion from a set of given premises. The **rules of replacement** allow for **logically equivalent** statements to be substituted axiomatically. That is, although statements may be constructed with different logical operators, they can express the same semantic meaning. Since these logically equivalent statements have been determined, and captured in the rules below, they can be used to replace one another in natural deduction. Notice that each rule of replacement establishes the logically equivalent statements by use of a double colon :: with the statements listed on either side. This means that the substitution of statements can go either way (from left to right, or right to left). It is worth noting that the rules of replacement can be used on entire premises or some constituent part of a compound premise. The rules of replacement take the symbolic forms below:

9. **DeMorgan's Rule (DM):**

$\sim(P \mathbin{\&} Q) :: (\sim P \vee \sim Q)$

$\sim(P \vee Q) :: (\sim P \mathbin{\&} \sim Q)$

Here are some additional examples of DeMorgan's rules applied to more complex statements:

$$\frac{P \longrightarrow \sim(Q \ \& \ R)}{P \longrightarrow (\sim Q \vee \sim R)}$$

$$\frac{\sim[(P \longrightarrow Q) \ \& \ (R \longrightarrow S)]}{\sim(P \longrightarrow Q) \vee \sim(R \longrightarrow S)}$$

$$\frac{[\sim(Q \longrightarrow R) \ \& \ \sim(S \longrightarrow T)] \longleftrightarrow \sim P}{\sim[(Q \longrightarrow R) \vee (S \longrightarrow T)] \longleftrightarrow \sim P}$$

10. Commutativity (Comm):

$(P \vee Q) :: (Q \vee P)$

$(P \ \& \ Q) :: (Q \ \& \ P)$

Here are some additional examples of Commutativity applied to more complex statements:

$$\frac{P \longrightarrow (Q \ \& \ R)}{P \longrightarrow (R \ \& \ Q)}$$

$$\frac{(P \longrightarrow Q) \vee (R \longrightarrow S)}{(R \longrightarrow S) \vee (P \longrightarrow Q)}$$

$$\frac{[P \longrightarrow (Q \ \& \ R)] \vee [S \longrightarrow (Q \ \& \ T)]}{[P \longrightarrow (Q \ \& \ R)] \vee [S \longrightarrow (T \ \& \ Q)]}$$

$$\frac{[P \longrightarrow (Q \ \& \ R)] \ \& \ [S \longrightarrow (Q \ \& \ T)]}{[S \longrightarrow (Q \ \& \ T)] \ \& \ [P \longrightarrow (Q \ \& \ R)]}$$

11. Associativity (Assoc):

$[P \vee (Q \vee R)] :: [(P \vee Q) \vee R]$

$[P \ \& \ (Q \ \& \ R)] :: [(P \ \& \ Q) \ \& \ R]$

12. Distribution (Dist):

$[P \ \& \ (Q \vee R)] :: [(P \ \& \ Q) \vee (P \ \& \ R)]$

$[P \vee (Q \ \& \ R)] :: [(P \vee Q) \ \& \ (P \vee R)]$

13. Double Negation (DN):

$P :: \sim\sim P$

Here is additional example of Double Negation applied to a more complex statement:

$$\frac{P \longrightarrow (Q \ \& \ R)}{\sim\sim P \longrightarrow (Q \ \& \ R)}$$

14. Transposition (Trans):

$(P \longrightarrow Q) :: (\sim Q \longrightarrow \sim P)$

Here are some additional examples of Transposition applied to more complex statements:

$$\frac{P \longrightarrow (Q \ \& \ R)}{\sim(Q \ \& \ R) \longrightarrow \sim P}$$

$$\frac{(P \longrightarrow Q) \vee (R \longrightarrow S)}{(\sim Q \longrightarrow \sim P) \vee (R \longrightarrow S)}$$

$$\frac{[P \longrightarrow (Q \ \& \ R)] \vee [S \longrightarrow (Q \ \& \ T)]}{[P \longrightarrow (Q \ \& \ R)] \vee [\sim(Q \ \& \ T) \longrightarrow \sim S]}$$

15. Material Implication (Impl):

(P⟶Q) :: (∼P∨Q)

Here are some additional examples of Material Implication applied to more complex statements:

$$\frac{\text{P}\longrightarrow(\text{Q \& R})}{\sim\text{P}\vee(\text{Q \& R})}$$

$$\frac{(\text{P}\longrightarrow\text{Q})\vee(\text{R}\longrightarrow\text{S})}{(\sim\text{P}\vee\text{Q})\vee(\text{R}\longrightarrow\text{S})}$$

$$\frac{\sim(\text{P}\longrightarrow\text{Q})\vee(\text{R}\longrightarrow\text{S})}{(\text{P}\longrightarrow\text{Q})\longrightarrow(\text{R}\longrightarrow\text{S})}$$

$$\frac{[\text{P}\longrightarrow(\text{Q \& R})]\vee[\text{S}\longrightarrow(\text{Q \& T})]}{[\text{P}\longrightarrow(\text{Q \& R})]\vee[\sim\text{S}\vee(\text{Q \& T})]}$$

16. Material Equivalence (Equiv):

(P⟷Q) :: (P⟶Q) & (Q⟶P)

(P⟷Q) :: (P & Q)∨(∼P & ∼Q)

17. Exportation (Exp):

[(P & Q)⟶R] :: [P⟶(Q⟶R)]

Here is additional example of Exportation applied to a more complex statement:

$$\frac{(\text{P \& Q})\longrightarrow(\text{R \& S})}{\text{P}\longrightarrow[\text{Q}\longrightarrow(\text{R \& S})]}$$

18. Tautology (Taut):

P :: P & P

P :: P∨P

Here are some additional examples of Tautology applied to more complex statements:

$$\frac{(\text{P \& Q})\ \&\ (\text{P \& Q})}{\text{P \& Q}}$$

$$\frac{[\text{P}\longrightarrow(\text{Q \& R})]\vee[\text{P}\longrightarrow(\text{Q \& R})]}{\text{P}\longrightarrow(\text{Q \& R})}$$

Helpful Hints for Rules of Inference

1. The rules of *implication* only apply to the main logical operators in each participating premise.

2. The rules of *replacement* can be used on entire premises or some constituent part of a compound premise.

3. If no obvious rules of replacement can be applied, the rules of implication can be used to set up for rules of replacement.

4. It is easier to use the rules of inference in natural deduction if you have memorized them. Another helpful way of remembering the rules is to know which ones apply to which logical operators (so as not to be overwhelmed by all of the possibilities). Below is a helpful guide about which rules involve which logical operator(s), or combinations thereof:

Logical Operator(s)	Rules of Inference
∼	Double Negation
&	Simplification, Conjunction, Commutativity, Associativity, Tautology
∨	Addition, Commutativity, Associativity, Tautology
⟶	Modus Ponens, Hypothetical Syllogism, Transposition
⟶ , ∼	Modus Tollens
∨ , ∼	Disjunctive Syllogism
⟶ , ∨	Constructive Dilemma, Material Implication
∨ , &	DeMorgan's, Distribution
⟶ , &	Exportation
⟷, ⟶, &	Material Equivalence

3.3 Derivations in Propositional Logic: Steps 1-5

1. Deduction is best approached like a puzzle: begin with the conclusion and work backwards to determine how to separate out the conclusion from the premises given. In order to begin deriving the conclusion from a set of given premises, look for the variables of the conclusion in the given premises. This will give some indication as to where to begin. Consider the following argument:

1. A & (B∨C)
2.∼A∨∼C / **A & B**

Notice that the conclusion here is set up across from the last premise and is demarcated as such by a /, this tells us what we need to derive from the premises given on the right-hand side. We can also see that both the variables of the conclusion occur in the first premise, and one of them occurs in the second premise.

2. Once the variables in the conclusion have been located in the premises, look for the main logical operators at use in those premises. This will give some indication as to which rules of inference to use in order to separate out the desired variables from their original locations.

1. **A** & (**B**∨C)
2.∼A∨∼C / A & B

Since both variables in the conclusion appear in the first premise, we will start there as a strategic move.

1. A & (B∨C)
2.∼A∨∼C / A & B
3. (A & B)∨(A & C)

If we review our guide for applying rules of inference, we know that there are only two rules which involve conjunctive and disjunctive statements (DeMorgan's and Distribution). However, we also know that DeMorgan's only applies to one of those operators at a time, so we have used Distribution to break apart the main conjunctive operator and imply what is now, line 3. The value of applying this rule is that it places our desired variables together as they appear in the conclusion.

3. When utilizing the rules of inference in natural deduction, be sure to identify which line(s) were used and include the abbreviation for the rule applied on the left-hand side, as illustrated below:

1. A & (B∨C)
2.∼A∨∼C / A & B
3. (A & B)∨(A & C) 1 (Dist)

4. Repeat Steps 2 and 3 using each derived line as an additional resource to arrive at the desired conclusion.

1. A & (B∨C)
2.∼A∨∼C / A & B
3. (A & B)∨(A & C) 1 (Dist)
4. ∼(A & C) 2 (DM)

Here, since there was nothing that could be done with our new third line at this time, we have replaced line 2 with its logical equivalent on what is now, line 4. We have also been sure to note the line and rule used.

1. A & (B∨C)	
2.∼A∨∼C	/ A & B
3. (A & B)∨(A & C)	1 (Dist)
4. ∼(A & C)	2 (DM)
5. A & B	3, 4 (DS)

Finally, we can see that lines 3 and 4 can work together to create a Disjunctive Syllogism, which gives us our conclusion on what is now, line 5. Notice here that since two lines were used in order to arrive at line 5, both lines are listed on the right as part of the justification.

5. A final check of the lines used on the right-hand side will let us know if there are any superfluous or unnecessary moves. It is worth noting that derivations can be done in different ways, using different rules or the same rules in a different order. The important thing is that **each rule we apply needs to be used in arriving at the conclusion**. It is also worth noting that lines may be used as many times as necessary in deriving the desired conclusion.

1. A & (B∨C)	
2.∼A∨∼C	/ A & B
3. (A & B)∨(A & C)	1 (Dist)
4. ∼(A & C)	2 (DM)
5. A & B	3, 4 (DS)

Here, we can see that there are no superfluous moves, since each line prior to the conclusion is used at least once.

The rules for natural deduction introduced thus far are part of what is called the **direct method**, which means that the conclusion is derived using only the premises given, as well as what can be directly implied by, or is logically equivalent to, those premises. In the following section, we will see what can be derived by combing the rules of inference with assumptions that go beyond what is given in the premises.

3.4 Conditional Proof

In some cases, the desired conclusion cannot be derived solely based on the premises given. Nor can it be derived by the logical equivalent or implication of those premises. In cases such as these (or others where, although it may be possible to use the direct method, it would be far more laborious), we may opt to use a less direct method for deriving the desired conclusion. This less direct method is particularly preferable when dealing with arguments whose conclusion is a conditional statement (or where a conditional is needed to derive the conclusion, but cannot be inferred from the premises given). For such arguments, the **conditional proof** method allows us to assume the antecedent of the conclusion in what we call a subderivation, derive the consequent of the conclusion, and then discharge the conditional statement implied by the subderivation.

A **subderivation** is a derivation within a derivation. The same rules of inference which apply in derivations also apply in subderivations, however, *any lines in the subderivation may not be used outside of that subderivation in the main derivation* (since they all arise from an assumption which has not been established by the given premises). Once the consequent of the conditional statement is derived from the assumed antecedent, the resulting conditional statement can be **discharged** from the subderivation and used in the main derivation, using the lines of the subderivation as its justification.

Conditional proofs take the symbolic form below:

| x. P	(ACP)
| y. ...	
| z. Q	
zz. P⟶Q	x-z (CP)

Notice here that the antecedent of the desired conditional statement is assumed on the first line of the subderivation. The abbreviation for the assumption is (ACP) for "assumption for conditional proof". The assumption, and all implied lines in the subderivation are demarcated by a line along the left-hand side. The subderivation is complete when the desired consequent has been derived from the assumption. Sometimes the desired consequent cannot be derived from the assumption alone, and so may need to be implied by the assumption and the given premises in the main derivation. Finally, the assumption becomes the antecedent of the resulting conditional statement, with the final line of the subderivation becoming the consequent of the conditional statement. The conditional statement has been discharged from the subderivation, and so is not inside the line on the left. The resulting conditional statement is justified by all lines in the subderivation (however many there may be), and the abbreviation for the rule used is (CP) for "conditional proof".

Here are some additional examples of Conditional Proof used to derive conditional statements in more complex arguments:

1. $P \longrightarrow (Q \,\&\, R)$	
2. $(Q \vee S) \longrightarrow T$	/ $P \longrightarrow T$
│ 3. P	(ACP)
│ 4. $Q \,\&\, R$	1, 3 (MP)
│ 5. Q	4 (Simp)
│ 6. $Q \vee S$	5 (Add)
│ 7. T	2, 6 (MP)
8. $P \longrightarrow T$	3-7 (CP)

1. $P \longrightarrow (Q \,\&\, R)$	
2. $S \longrightarrow (T \,\&\, U)$	
3. $P \vee S$	/ $Q \vee T$
│ 4. P	(ACP)
│ 5. $Q \,\&\, R$	1, 4 (MP)
│ 6. Q	5 (Simp)
7. $P \longrightarrow Q$	4-6 (CP)
│ 8. S	(ACP)
│ 9. $T \,\&\, U$	2, 8 (MP)
│ 10. T	5 (Simp)
11. $S \longrightarrow T$	8-10 (CP)
12. $(P \longrightarrow Q) \,\&\, (S \longrightarrow T)$	7, 11 (Conj)
13. $Q \vee T$	3, 12 (CD)

1. $P \longrightarrow [Q \longrightarrow (R \vee S)]$	/ $P \longrightarrow (\sim Q \vee S)$
2. $Q \longrightarrow \sim R$	
│ 3. P	(ACP)
│ 4. $Q \longrightarrow (R \vee S)$	1, 3 (MP)
│ │ 5. Q	(ACP)
│ │ 6. $R \vee S$	4, 5 (MP)
│ │ 7. $\sim R$	2, 5 (MP)
│ │ 8. S	6, 7 (DS)
│ 9. $Q \longrightarrow S$	5-8 (CP)
│ 10. $\sim Q \vee S$	9 (Impl)
11. $P \longrightarrow (\sim Q \vee S)$	3-10 (CP)

3.5 Indirect Proof

Similar to the conditional proof method, introduced in the previous section, we may need to derive a negated conclusion, but are unable to do so from the premises given (or doing so would require far too lengthy a derivation). In such cases, the **indirect proof** method allows us to assume the opposite from what we hope to derive in the conclusion, use the opposite of that negated conclusion to derive a logical contradiction, and then discharge the desired conclusion. This method of is similar to the argument strategy commonly known as ***reductio ad absurdum*** where we assume the opposite of what we are trying to prove, and then use that assumption to show that if our conclusion is rejected, it implies something absurd (or in this case, logically contradictory). When using the indirect proof method in natural deduction, the logical contradiction will take the form of deriving two statements, one of which is the negation of the other, and then conjoining those statements to illustrate the contradiction.

Indirect proofs take the symbolic form below:

```
  | x.∼P           (AIP)
  |-------------------------
  | y. ...
  | z. Q
  | zz.∼Q
  | zy. Q &∼Q     z, zz (Conj)
zx. P                x-zy (IP)
```

Notice here that the negation of the desired statement is assumed. The abbreviation for the assumption is (AIP) for "assumption for indirect proof". The assumption, and all implied lines in the subderivation are demarcated by a line along the left-hand side. The subderivation is complete when the assumption is used to imply two contradictory statements. Sometimes a logical contradiction cannot be derived from the assumption alone, and so may need to be implied by the assumption and the given premises in the main derivation. Finally, the contradictory statements are conjoined on the final line of the subderivation, allowing the opposite of the assumption to be discharged. The resulting statement is justified by all lines in the subderivation (however many there may be), and the abbreviation for the rule used is (IP) for "indirect proof".

Here are some additional examples of Indirect Proof used to derive negated statements in more complex arguments:

```
1. (P∨Q)⟶(R & S)
2. R⟶∼S                / ∼P
  | 3. P                 (AIP)
  |--------------------------------
  | 4. P∨Q               3 (Add)
  | 5. R & S             1, 3 (MP)
  | 6. R                 5 (Simp)
  | 7. ∼S                2, 6 (MP)
  | 8. S                 5 (Simp)
  | 9. S &∼S             7, 8 (Conj)
8. ∼P                   3-9 (IP)
```

```
1. P⟶[(Q∨R)⟶(S & T)]
2. P & ∼(T∨U)               / ∼(Q∨U)
3. P                         2 (Simp)
4. (Q∨R)⟶(S & T)            1, 3 (MP)
5.∼(T∨U)                     2 (Simp)
6.∼T &∼U                     5 (DM)
  | 7. Q                       (AIP)
  |------------------------------------
  | 8. Q∨R                     7 (Add)
  | 9. S & T                   4, 8 (MP)
  | 10. T                      9 (Simp)
  | 11. ∼T                     6 (Simp)
  | 12. T &∼T                  10, 11 (Conj)
13.∼Q                        7-12 (IP)
14.∼U                        6 (Simp)
15.∼Q &∼U                    13, 14 (Conj)
16.∼(Q∨U)                    15 (DM)
```

1. P⟶[∼Q⟶(R & S)]	
2.∼R & T	/ P⟶(Q & T)
3. P	(ACP)
4.∼Q⟶(R & S)	1, 3 (MP)
5. ∼Q	(AIP)
6. R & S	4, 5 (MP)
7. R	6 (Simp)
8. ∼R	2 (Simp)
9. R &∼R	7, 8 (Conj)
10.∼∼Q	5-9 (IP)
11. Q	10 (DN)
12. T	2 (Simp)
13. Q & T	11, 12 (Conj)
14. P⟶(Q & T)	3-13 (CP)

3.6 Proving Theorems

In additional to being able to prove that a conclusion can be derived from its premises, as we covered in the previous sections of this chapter, it is also important to know how to approach statements that may be used like a conclusion, but which are presented without any supporting premises. This is also helpful when constructing an argument of one's own, since we often know first what we would like to argue for, but have not yet figured out a deductively valid way of doing so. For these instances, as well as many others often presented in math and theoretical physics, we would opt to use both the indirect and conditional proof methods to prove our desired conclusion. When a statement is proven using only the rules of inference through either the indirect and/or conditional proof methods, we call these statements, theorems. **Theorems** are axiomatic statements which are proven solely on the basis of previously established statements (in our case, the rules of inference, conditional proofs, and indirect proofs). As such, they can be established without the presence of any given premises.

Proving Theorems: Steps 1-3

1. Determine which method should be used first (conditional or indirect). As we can see with a quick review of the indirect and conditional proof methods, provided below, the most appropriate method will depend on the logical structure of the theorem being proven. If the theorem has a conditional as its main logical operator, the conditional proof method is most appropriate. If the theorem does not have a conditional as its main logical operator, then the indirect proof method is most appropriate. However, it is worth noting that either method could be used for proving any theorem. Although, in many cases, one particular method will be more strategic.

Conditional Proof Form	**Indirect Proof Form**
x. P (ACP) y. ... z. Q zz. P⟶Q x-z (CP)	x.∼P (AIP) y. ... z. Q zz.∼Q zy. Q &∼Q z, zz (Conj) zx. P x-zy (IP)

2. Determine the assumption for either the conditional or indirect proof method being used. This assumption will act as the first line in the derivation, giving you something to work with in order to begin proving the desired theorem.

In theorem proofs for simpler compound statements, the logical form of the method being used will determine what will be assumed. If the conditional proof method is being used, the assumption will be the antecedent condition of the theorem being proven. If the indirect proof method is being used, the assumption will be the negation of the theorem being proven. Consider the following examples of the same theorem being proven using each method:

/ P⟶(Q⟶P)
1. P (ACP)
...

/ P⟶(Q⟶P)
1.∼[P⟶(Q⟶P)] (AIP)
...

Additionally, here is an example with a more complex conditional statement:

/ [P⟶(Q⟶R)]⟶[(P⟶Q)⟶(P⟶R)]
| 1. P⟶(Q⟶R) (ACP)
| ...

In theorem proofs for statements of **logical equivalence**, the derivation will always utilize two conditional proofs. The antecedent of the theorem is assumed in the first conditional proof and used to derive the consequent of the equivalent statement. The consequent of the theorem is assumed in the second conditional proof and used to derive the antecedent of the equivalent statement. The resulting discharged conditional statements are then conjoined and the Equivalence rule of replacement is used to arrive at the desired theorem. This approach takes the symbolic form below:

/ P⟷Q
| 1. P (ACP)
| ...
| x. Q
y. P⟶Q 1-x (CP)
| z. Q (ACP)
| ...
| zz. P
zy. Q⟶P z-zz (CP)
zx. (P⟶Q) & (Q⟶P) y, zy (Conj)
yz. P⟷Q zx (Equiv)

When applied to the following example, we would use the assumptions illustrated below:

/ P⟷P & (Q⟶P)
| 1. P (ACP)
| ...
x. ...
| y. P & (Q⟶P) (ACP)
| ...

3. Once the method and assumption have been determined, the rules of inference may be applied as necessary in order to derive the desired theorem. Using the examples above, we can complete each derivation as follows:

/ P⟶(Q⟶P)
| 1. P (ACP)
| | 2. Q (ACP)
| | 3. P∨P 1 (Add)
| | 4. P 3 (Taut)
| 5. Q⟶P 2-4 (CP)
6. P⟶(Q⟶P) 1-5 (CP)

/ P⟶(Q⟶P)
| 1.∼[P⟶(Q⟶P)] (AIP)
| 2.∼[P∨(Q⟶P)] 1 (Impl)
| 3.∼[P∨(∼Q∨P)] 2 (Impl)
| 4.∼∼P &∼(∼Q∨P) 3 (DM)
| 5. P &∼(∼Q∨P) 4 (DN)
| 6. P & (∼∼Q &∼P) 5 (DM)
| 7. P & (∼P &∼∼Q) 6 (Comm)
| 8. (P &∼P) &∼∼Q 7 (Assoc)
| 9. P &∼P 8 (Simp)
10. ∼∼[P⟶(Q⟶P)] 1-9 (IP)
11. P⟶(Q⟶P) 10 (DN)

/ $[P \longrightarrow (Q \longrightarrow R)] \longrightarrow [(P \longrightarrow Q) \longrightarrow (P \longrightarrow R)]$

1. $P \longrightarrow (Q \longrightarrow R)$	(ACP)
2. $P \longrightarrow Q$	(ACP)
3. P	(ACP)
4. $Q \longrightarrow R$	1,3 (MP)
5. Q	2, 3 (MP)
6. R	4, 5 (MP)
7. $P \longrightarrow R$	3-6 (CP)
8. $(P \longrightarrow Q) \longrightarrow (P \longrightarrow R)$	2-7 (CP)
9. $[P \longrightarrow (Q \longrightarrow R)] \longrightarrow [(P \longrightarrow Q) \longrightarrow (P \longrightarrow R)]$	1-8 (CP)

/ $P \longleftrightarrow P \ \& \ (Q \longrightarrow P)$

1. P	(ACP)
2. $P \vee \sim Q$	1 (Add)
3. $\sim Q \vee P$	2 (Comm)
4. $Q \longrightarrow P$	3 (Impl)
5. $P \ \& \ (Q \longrightarrow P)$	1, 4 (Conj)
6. $P \longrightarrow P \ \& \ (Q \longrightarrow P)$	1-5 (CP)
7. $P \ \& \ (Q \longrightarrow P)$	(ACP)
8. P	7 (Simp)
9. $P \ \& \ (Q \longrightarrow P) \longrightarrow P$	7-8 (CP)
10. $[P \longrightarrow P \ \& \ (Q \longrightarrow P)] \ \& \ [P \ \& \ (Q \longrightarrow P) \longrightarrow P]$	6, 9 (Conj)
11. $P \longleftrightarrow P \ \& \ (Q \longrightarrow P)$	10 (Equiv)

3.7 An Overview of Rules for Propositional Logic

Rules of Inference

Rules of Implication	
Modus Ponens (MP) $\begin{array}{l} P\longrightarrow Q \\ P \\ \hline Q \end{array}$	**Modus Tollens (MT)** $\begin{array}{l} P\longrightarrow Q \\ \sim Q \\ \hline \sim P \end{array}$
Hypothetical Syllogism (HS) $\begin{array}{l} P\longrightarrow Q \\ Q\longrightarrow R \\ \hline P\longrightarrow R \end{array}$	**Disjunctive Syllogism (DS)** $\begin{array}{l} P\lor Q \\ \sim Q \\ \hline P \end{array}$ $\quad\begin{array}{l} P\lor Q \\ \sim P \\ \hline Q \end{array}$
Constructive Dilemma (CD) $\begin{array}{l} (P\longrightarrow Q)\ \&\ (R\longrightarrow S) \\ P \lor R \\ \hline Q \lor S \end{array}$	**Simplification (Simp)** $\begin{array}{l} P\ \&\ Q \\ \hline P \end{array}$ $\quad\begin{array}{l} P\ \&\ Q \\ \hline Q \end{array}$
Conjunction (Conj) $\begin{array}{l} P \\ Q \\ \hline P\ \&\ Q \end{array}$	**Addition (Add)** $\begin{array}{l} P \\ \hline P\lor Q \end{array}$

Rules of Replacement	
DeMorgan's (DM) $\sim(P\ \&\ Q) :: (\sim P\lor\sim Q)$ $\sim(P\lor Q) :: (\sim P\ \&\ \sim Q)$	**Commutativity (Comm)** $(P\lor Q) :: (Q\lor P)$ $(P\ \&\ Q) :: (Q\ \&\ P)$
Associativity (Assoc) $[P\lor(Q\lor R)] :: [(P\lor Q)\lor R]$ $[P\ \&\ (Q\ \&\ R)] :: [(P\ \&\ Q)\ \&\ R]$	**Distribution (Dist)** $[P\ \&\ (Q\lor R)] :: [(P\ \&\ Q)\lor(P\ \&\ R)]$ $[P\lor(Q\ \&\ R)] :: [(P\lor Q)\ \&\ (P\lor R)]$
Double Negation (DN) $P :: \sim\sim P$	**Transposition (Trans)** $(P\longrightarrow Q) :: (\sim Q\longrightarrow\sim P)$
Material Implication (Impl) $(P\longrightarrow Q) :: (\sim P\lor Q)$	**Exportation (Exp)** $[(P\ \&\ Q)\longrightarrow R] :: [P\longrightarrow(Q\longrightarrow R)]$
Material Equivalence (Equiv) $(P\longleftrightarrow Q) :: (P\longrightarrow Q)\ \&\ (Q\longrightarrow P)$ $(P\longleftrightarrow Q) :: (P\ \&\ Q)\lor(\sim P\ \&\ \sim Q)$	**Tautology (Taut)** $P :: P\ \&\ P$ $P :: P\lor P$

Conditional Proof Form	**Indirect Proof Form**
| x. P (ACP) | y. ... | z. Q zz. $P\longrightarrow Q$ x-z (CP)	| x. $\sim P$ (AIP) | y. ... | z. Q | zz. $\sim Q$ | zy. $Q\ \&\sim Q$ z, zz (Conj) zx. P x-zy (IP)

Chapter 4

Predicate Logic with Natural Deduction

4.1 Translation and Symbols for Predicate Logic

In Chapter Two, we noted that even the most complex statements in ordinary language can be translated into symbolic form. However, there are many cases in which we might want, or need, to symbolize the statements we are dealing with in ways that capture more of the semantic subtleties of the statement's meaning. In **predicate logic**, variables are used to symbolize both the subject and predicate of each simple statement (rather than representing one or the other, as is done in propositional logic). As with propositional logic though, these additional variables can also be used to capture even the most complex compound statements. Let us briefly introduce some examples of the difference between the symbols in propositional logic and what we will learn in predicate logic:

Statement	Propositional Logic:	Predicate Logic:
The sky is blue.	S	Bs
The sky is blue and cloudy.	B & C	Bs & Cs
The sky is blue and sunny.	B & S	Bs & Ss

Notice here that the variables in predicate logic are better able to represent their subject and predicate counterparts since both receive their own variables.

Along with more accurately capturing the relationship between each statement's predicate and corresponding subject, there may be additional types of statements that we would like to translate into symbolic form. However, sometimes those statements have a semantic meaning that cannot be adequately captured by the tools of propositional logic. For example, let us briefly consider how the following statements would be translated in propositional logic:

Statement	Translation
All philosophers love logic.	P
Most philosophers love logic.	P
A philosopher loves logic.	P

It should be immediately obvious that this cannot be adequate, since all three statements have very different meanings, and yet are being represented by the same variable.

In order to properly capture the difference in semantic meaning between these types of statements, we will need additional symbols and rules which deal with categorical statements. **Categorical statements** are those that establish a relation between the class of the subject and the participation of that class in the predicate. The class may refer to either a part or the whole of the statement's subject(s), and may either be included or excluded to some degree in the statement's predicate(s).

There are four types of categorical statements:

Form	Category
All S are P.	Whole subject (every member) is included in predicate.
Some S are P.	Part of the subject (at least one) is included in predicate.
No S are P.	Whole subject (every member) is excluded from predicate.
Some S are not P.	Part of the subject (at least one) is excluded from predicate.

Notice here that the variables 'S' and 'P' are used to represent the subject and predicate of each statement, respectively.

Symbols

When reproducing simple statements in predicate logic, new variables are needed in order to better capture the relationship between the various subject(s) and predicate(s) within a given statement. Rather than only selecting one variable to represent each simple statement, predicate logic represents both the subject and predicate of each statement with their own symbols. **Predicate variables** of all 26 uppercase letters of the Roman alphabet (A, B, C, D,. . . W, X, Y, Z) are used that best represent the semantic meaning of the statement's *predicate*. Examples of predicate variables can be seen below:

Statement	Predicate Variable
It is a rainy day	R
I need an umbrella	U
Anthony has five arms	F
Anthony is a philosopher	P

Although the predicate variable is a major part of what separates predicate from propositional logic, this variable alone is not complete.

Along with symbols for predicates, **constants** of *only* the first 23 lowercase letters of the Roman alphabet (a, b, c, d, . . . t, u, v, w) are used to best represent the semantic meaning of the statement's *subject*. Constants are assigned to the right-hand side of their corresponding predicate variables, as illustrated below:

Statement	Constant Variable [Subject]
It is a rainy day	Rd
I need an umbrella	Ui
Anthony has five arms	Fa
Anthony is a philosopher	Pa

Notice that in some cases, more than one simple statement may have the same subject (as in the last two examples) or same predicate, while in others, the subject may not be explicit. When various simple statements in the same argument have the same subject or predicate reference, the same variable can be used. However, if two or more different subjects or predicates begin with the same letter, different variables need to be selected to reflect their different semantic meaning, as illustrated below:

Statement	Variable Selection
Anthony has five arms	Fa
Anthony is a philosopher	Pa
Rebeka is a philosopher	Pr
Rebeka has five arms	Fr

Statement	Variable Selection
Albert has five arms	Fa
Anthony is a philosopher	Pt
Rebeka is a psychologist	Sr
Rachel has five cars	Cl

Notice how the first set of examples repeat subject and predicate variables since the subjects and predicates have the same semantic meaning in each statement. However, in the second set of examples, different variables are used because there are different subjects and predicates in each statement ("Albert" and "Anthony" cannot both be represented by the constant 'a', "philosophy" and "psychology" cannot both be represented by the predicate variable 'P', etc.)

When reproducing categorical statements in predicate logic, new symbols are needed in order to capture the quantity of subject classes participating in the predicate class, as well as the relationship between the statement's subject(s) and predicate(s). The four categorical statements introduced earlier in this chapter can be separated into two broader categories which capture the quantity expressed in the statement. Appropriately, each of these larger categories will be represented by what are called **quantifiers**. There are quantifiers for universal and existential statements. **Universal statements** capture the universal exclusion or inclusion of a subject class in a predicate class, while **existential statements** capture the particular exclusion or inclusion of a subject class in a predicate class. The universal and existential quantifiers are represented by the following symbols, respectively:

Quantifier	Categorical Statements		Symbol
Universal[1]	Affirmative	All S are P.	(x)
	Negative	No S are P.	∼(x)
Existential[2]	Affirmative	Some S are P.	(∃x)
	Negative	Some S are not P.	(∃x)∼

Notice how the variable 'x' is used as part of the symbolization of each categorical statement above. The last three lowercase letters of the Roman alphabet (x, y, and z) are **quantifier variables** in that they are always paired with universal and existential quantifiers. Quantifier variables are always used in sequential order, unless one or more of the previous quantifier variables have already been assigned. It is worth noting that the designation of these three letters as quantifier variables is why only the first 23 lowercase letters of the Roman alphabet can be used to represent constants.

While the categorical designation is the last big piece of symbolization in predicate logic, they are never complete without being assigned to a simple or compound statement.

Each quantifier will be assigned to the left-hand side of a simple or compound statement. However, as soon as this assignment is made, the statement needs to be reinterpreted to adequately account for the relationship between the members of the subject class and the extent of their participation in the predicate class. Thus, the assignment of quantifiers is best illustrated in the subsequent section on translation.

Translation

We can see above that *simple statements* in predicate logic are translated using a single predicate variable accompanied by a single corresponding constant. This means that each simple statement requires only one of each in order to adequately capture the statement's subject and predicate.

For compound statements in predicate logic, the predicate variables and corresponding constants are connected using the same logical operators as those introduced in propositional logic. Similarly then, the negation operator can be attached to a single simple statement while all other operators are used to connect two simple statements. For example:

Statement	Translation
It is not the case that Anthony has five arms.	∼Fa
Both Anthony and Rebeka are philosophers.	Pa & Pr
Rebeka is either a philosopher, or a psychologist.	Pr∨Sr
If Rebeka has five arms, then Rachel has five cars.	Fr⟶Cl
Anthony will go to the movies, if an only if Rebeka goes.	Ma⟷Mr

For more *complex compound statements*, parentheses and brackets are used to demarcate various subcomponents of each statement, as in propositional logic. Let us compare another example translated in both propositional and predicate logic:

If it is a rainy day, then Anthony will go the movies and Rebeka will stay home.

Translation	Propositional Logic:	Predicate Logic:
	R⟶(A & H)	Rd⟶(Ma & Hr)

Notice here that in propositional logic, since both "rain" and "Rebeka" begin with the letter 'R', different variables would be used to denote the semantic difference (in one case capturing the predicate of the weather and in the other, capturing the predicate of Rebeka's activity). In predicate logic, not only are different variables chosen to reflect the different predicates (rain, going to the movies, staying at home), we could actually use the same letter for both "rain" and "Rebeka" since the former is a predicate (represented by 'R') and the latter a constant (represented by 'r'). Thus, both could use the letter that best represents their semantic meaning because they were demarcated by the upper- and lower-case variables respectively.

For *categorical statements*, we must first understand the semantic meaning and symbolization behind each category in order to properly translate statements into predicate logic. Our system will understand the four categorical statements as follows:

Quantifier:	Universal Affirmative (x)
Statement:	All S are P.
Semantic Meaning:	For all x, if x is S, then x is P.
Symbolization:	(x)(Sx⟶Px)

Quantifier:	Universal Negative ∼(x)
Statement:	No S are P.
Semantic Meaning:	For all x, if x is S, then x is not P.
Symbolization:	(x)(Sx⟶∼Px)

Notice here that universal statements are interpreted as meaning "for all x", and are then expressed in a conditional statement. Notice that the subject is always the antecedent, denoting the membership to this class as a necessary condition to being a member of the predicate class, which is always the consequent.

Quantifier:	Existential Affirmative (∃x)
Statement:	Some S are P.
Semantic Meaning:	There exists an x, such that x is S, and x is P.
Symbolization:	(∃x)(Sx & Px)

Quantifier:	Existential Negative (∃x)∼
Statement:	Some S are not P.
Semantic Meaning:	There exists an x, such that x is S, and x is not P.
Symbolization:	(∃x)(Sx &∼Px)

Notice here that existential statements are interpreted as meaning "there exists an x", and are then expressed in a conjunctive statement.

Once a categorical statement has been translated into predicate logic, we should be able to identify the **statement function**, that part of the symbolized statement that appears to the right of the quantifier. This statement function will be important when we later learn how to apply rules of inference and change of quantifier rules to predicate logic in natural deduction.

Helpful Hints for Translation in Predicate Logic

Noncategorical and Categorical Statements

Noncategorical statements can be interpreted either as singular or compound statements, or as existential statements. It is also worth noting that whenever we symbolize statements categorically, this causes the variable(s) to change from what they might otherwise be in a noncategorical statement. Consider the following examples:

Category	Phrasing	Symbolization
Noncategorical	It is raining	R
Existential	It is a rainy day	(∃x)(Dx & Rx)
Universal	Every day is a rainy day	(x)(Dx⟶Rx)

Category	Phrasing	Symbolization
Noncategorical	It is not raining	∼R
Existential	It is not a rainy day	(∃x)(Dx & ∼Rx)
Universal	No days are rainy days	(x)(Dx⟶∼Rx)

Notice here that what was the only variable needed to symbolize the noncategorical statement became the predicate variable to the constant (x) in the existential and universal statements. Additionally, notice how the categorical phrasing of the existential statement is rephrased. Even though both the existential and noncategorical statements refer to a single instance, the noncategorical statement lacks the necessary subject and predicate to be translated into predicate logic. Finally, the existential statement is symbolized using the conjunctive interpretation, and the universal statement is symbolized using the conditional interpretation.

Complex Categorical Statements

For categorical statements in predicate logic, the rules of translation outlined above, are merely combined with the universal and existential quantifier symbols introduced in this chapter. However, since the translation of categorical statements can be lengthy, and this length can be multiplied by having more than one quantifier in a single proposition, we have constructed a quick guide for reference when translating complex categorical statements:

Negated Categorical Statements	**Translation**
It is not the case that all S are P.	∼(x)(Sx⟶Px)
It is not the case that no S are P.	∼[(x)(Sx⟶∼Px)]
It is not the case that some S are P.	∼[(∃x)(Sx & Px)]
It is not the case that some S are not P.	∼[(∃x)(Sx &∼Px)]

Conjunctive Categorical Statements	**Translation**
All S are P, and some S are P.	[(x)(Sx⟶Px)] & [(∃x)(Sx & Px)]
Some S are P and some S are not P.	[(∃x)(Sx & Px)] & [(∃x)(Sx &∼Px)]

Disjunctive Categorical Statements	**Translation**
Either all S are P, or no S are P.	[(x)(Sx⟶Px)]∨[(x)(Sx⟶∼Px)]
Either some S are P, or some S are not P.	[(∃x)(Sx & Px)]∨[(∃x)(Sx &∼Px)]

Conditional Categorical Statements	**Translation**
If all S are P, then some S are P.	[(x)(Sx⟶Px)]⟶[(∃x)(Sx & Px)]
If no S are P, then some S are not P.	[(x)(Sx⟶∼Px)]⟶[(∃x)(Sx &∼Px)]

Biconditional Categorical Statements	**Translation**
Some S are P, if and only if all S are P.	[(∃x)(Sx & Px)]⟷[(x)(Sx⟶Px)]
Some S are not P, if and only if no S are P.	[(∃x)(Sx &∼Px)]⟷[(x)(Sx⟶∼Px)]

Please note the reference guides above are not exhaustive of all of the different possible ways in which the four categorical statements can be connected in complex statements. They merely provide an illustration of what such arrangements will look like when translated into predicate logic.

In addition to predicate logic requiring new and different symbols from those used in propositional logic, some additional points are worth noting about translating categorical statements from ordinary language into logical symbolic form. Although the logical operators and basic rules of translation remain the same, the addition of quantifiers and multiplicity of ways in which categorical statements can be phrased often provide a bit of a challenge.

For instance, not all categorical statements will explicitly use quantifying phrases (such as "all", "some", or "none") and many may use verbage that needs to be reinterpreted before translating. In order to assist with these less obvious instances of categorical logic, as well instances that are more explicit, the following steps can be used to translate even the most complex categorical statements into predicate logic.

Translation in Predicate Logic: Steps 1-6

1. Determine whether or not you are dealing with a simple or compound statement (as one would do in propositional logic). This will determine whether or not you will be using any logical operators.

In order to illustrate the difference between translating noncategorical and categorical statements in predicate logic, we will consider the following two examples for each step:

Example 1:

If tomorrow is a snow day then Anthony will go the movies, and Rebeka will stay home if she has work to do and doesn't feel like a movie.

Example 2:

If tomorrow is a snow day then all of the students will stay home from school, and Rebeka will go to the movies if she does not have any work to do.

These are both complex compound statements.

2. For each simple statement, select the most appropriate predicate variable and corresponding constant. A helpful trick is to list the variables in the order that they appear with space in between to add their connectives, as illustrated below:

Example 1:

Variables:		Sd		Ma		
Statement:	If	tomorrow is a snow day	then	Anthony will go to the movies,		
	and	Rebeka will stay home	if	she has work to do	and	doesn't feel like a movie.
Variables:		Hr		Wr		Mr

Example 2:

Variables:		Sd		Hs
Statement:	If	tomorrow is a snow day	then	all of the students will stay home from school,
	and	Rebeka will go to the movies	if	she does not have any work to do.
Variables:		Mr		Wr

When laid out for translation into symbolic form:

Example 1:

Variables	**Translation**				
Sd, Ma, Hr, Wr, Mr	Sd	Ma	Hr	Wr	Mr

Example 2:

Variables	**Translation**			
Sd, Hs, Mr, Wr	Sd	Hs	Mr	Wr

3. In addition to determining whether you are dealing with a simple or compound statement, in predicate logic it is also necessary to determine whether each of the simple or compound statements is categorical. Universal statements assert that all or none of the members of the subject class are members of the predicate class (they are either all included or all excluded). Existential statements assert that one or more members of the subject class are members of the predicate class (at least one is included or excluded). If any of the simple or compound statements are categorical, determine whether that category is a universal affirmative $(x)(Sx \longrightarrow Px)$, universal negative $(x)(Sx \longrightarrow \sim Px)$, existential affirmative $(\exists x)(Sx \,\&\, Px)$, or existential negative $(\exists x)(Sx \,\& \sim Px)$. Returning to our example, we can locate the following demarcations of category:

Example 1:

Categories:		existential affirmative		existential affirmative		
Statement:	If	tomorrow is a snow day	then	Anthony will go to the movies,		
	and	Rebeka will stay home	if	she has work to do	and	doesn't feel like a movie.
Categories:		existential affirmative		existential affirmative		existential negation

Notice that in example one, all of the simple statements are about particular instances. This means that they could all be symbolized using the existential quantifier. However, since individual existential statements can be symbolized with the predicate variable and constant, and there are no simple statements present that involve categories of more than one, no quantifiers need be assigned.

Example 2:

Categories:		existential affirmative		universal affirmative
Statement:	If	tomorrow is a snow day	then	**all** of the students will stay home from school,
	and	Rebeka will go to the movies	if	she does **not** have **any** work to do.
Categories:		existential affirmative		universal negation

Notice that in the second example, even though many of the simple statements are again about particular instances, since at least one simple statement involves a category of more than one, quantifiers will be assigned to every simple statement.

4. For each categorical statement, assign the appropriate quantifier to the left-hand side of the statement reinterpreting the statement function appropriately, as illustrated below:

Example 2:

Categories:		(∃x)(Dx & Sx)		(x)(Tx⟶Cx)
Statement:	If	tomorrow is a snow day	then	all of the students will stay home from school,
	and	Rebeka will go to the movies	if	she does not have any work to do.
Categories:		(∃x)(Rx & Mx)		(x)(Rx⟶∼Wx)

Notice here that once each simple statement is translated as a categorical statement and reinterpreted accordingly, different variables needed to be chosen in order to avoid confusing the different semantic meaning of each simple statement's subject and predicate. A helpful note is, **if you know right from the beginning that you will be translating a categorical statement, begin by selecting variables for the categorical symbolization**, as we did here in Step 4, rather than selecting predicate variables and constants, as we did in Step 2. This will save a lot of time and prevent unnecessary steps.

When laid out for translation into symbolic form:

Example 2:

Variables	**Translation**			
D, S, T, C, R, M, W	(∃x)(Tx & Sx)	(x)(Tx⟶Cx)	(∃x)(Rx & Mx)	(x)(Rx⟶∼Wx)

5. For each compound statement, determine which logical operator will be used to connect each constituent simple statements. Be sure to also capture any negations in the statement.

Example 1:

Logical Operators:			⟶			
Statement:	If	tomorrow is a snow day	then	Anthony will go to the movies,		
	and	Rebeka will stay home	if	she has work to do	and	doesn't feel like a movie.
Logical Operators:	&		⟶		&	∼

When inserted into the translation into symbolic form:

Example 1:

Translation								
Sd	⟶	Ma	&	Hr	⟶	Wr	&	∼Mr

Example 2:

Logical Operators:			⟶	
Statement:	If	tomorrow is a snow day	then	all of the students will stay home from school,
	and	Rebeka will go to the movies	if	she does not have any work to do.
Logical Operators:	&		⟶	∼

When inserted into the translation into symbolic form:

Example 2:

Translation						
(∃x)(Dx & Sx)	⟶	(x)(Tx⟶Cx)	&	(∃x)(Rx & Mx)	⟶	(x)(Rx⟶∼Wx)

Notice that in both examples, conditionals, conjunctions, and negations have been added. However, as we saw with our examples for translation in propositional logic, we have a problem with the second conditional in both statements where the antecedents and consequent needs to be switched, as illustrated below:

Example 1:

Translation								
Sd	⟶	Ma	&	**Wr**	&	**∼Mr**	⟶	**Hr**

Example 2:

Translation						
(∃x)(Dx & Sx)	⟶	(x)(Tx⟶Cx)	&	**(x)(Rx⟶∼Wx)**	⟶	**(∃x)(Rx & Mx)**

Here the symbolized simple and categorical statements have been rearranged to correctly reflect the semantic meaning of the each proposition.

6. For complex compound statements, identify the main logical operator and place parentheses (and if needed, brackets) around each constituent statement. Be sure to identify the main logical operator and ensure that it is placed outside all parentheses and brackets, with the most secondary operators placed within the parentheses. Recall that the main logical operator will most often be placed near the semantically meaningful punctuation of the statement. Remember to begin separating the main logical operator from the secondary operators with parentheses, and then move on to brackets.

Example 1:

Statement:	If	tomorrow is a snow day	then	Anthony will go to the movies,		
	and	Rebeka will stay home	if	she has work to do	and	doesn't feel like a movie.
Main Operator:	&					

Notice here that we have begun by identifying the first conjunction as the main logical operator since it appears closest to the the punctuation in the statement (the comma). Since two sets of variables appear on the left side of the main logical operator, those can be placed in parentheses. However, we also know that the three remaining sets of variables on the right side of the main logical operator cannot all be in parentheses, so they will be placed in brackets, as illustrated below:

Example 1:

Translation								
(Sd	⟶	Ma)	&	[Wr	&	∼Mr	⟶	Hr]

Recall that what remains in the brackets will need to be broken up further, since as we have learned, operators can at most connect two sets of variables in predicate logic. This means that we will need to add parentheses around the most secondary compound statement inside the brackets. This change in notation is illustrated below:

Example 1:

Translation								
(Sd	⟶	Ma)	&	[(Wr	&	∼Mr)	⟶	Hr]

Notice here that since the variable sets 'Wr' and 'not-Mr' form the antecedent of this secondary conditional statement, they will be placed together within the parentheses. As the consequent of this secondary conditional statement, the variable set 'Hr' will go outside of the parentheses, but remain inside of the brackets.

Example 2:

Statement:	If	tomorrow is a snow day	then	all of the students will stay home from school,
	and	Rebeka will go to the movies	if	she does not have any work to do.
Main Operator:	&			

Similar to example 1, the conjunction is the main logical operator since it appears closest to the the punctuation in the statement (the comma). In this case however, parentheses are already present in the symbolization of the categorical statements, so we will begin with brackets. An important note here is that **brackets need to be placed around each categorical statement in a complex statement** in order to show that their respective statement functions are assigned to the corresponding quantifier. This notation is illustrated below:

Example 2:

Translation						
[(∃x)(Dx & Sx)]	⟶	[(x)(Tx⟶Cx)]	&	[(x)(Rx⟶∼Wx)]	⟶	[(∃x)(Rx & Mx)]

Since parentheses and brackets have now been utilized, we will need a third type of notation, braces { }, to demarcate the two conjuncts of the main logical operator. Since two categorical statements appear on either side of the main logical operator, each set of two can be placed inside the appropriate notation. This notation is illustrated below:

Example 2:

Translation						
{[(∃x)(Dx & Sx)]	⟶	[(x)(Tx⟶Cx)]}	&	{[(x)(Rx⟶∼Wx)]	⟶	[(∃x)(Rx & Mx)]}

4.2 Rules of Inference for Predicate Logic

All eighteen rules of inference, introduced in the first two sections of Chapter Three, can be used in predicate logic, with certain restrictions. In order to understand these restrictions, we must understand the relationship between the statement function and quantifier of both universal and existential statements. Recall that the statement function refers to the symbolization to the right of the quantifier. When quantifiers have simple statement functions, parentheses will only appear around the quantifier. When quantifiers have compound statement functions, the statement function will be placed within a second set of parentheses. So, for the universal statement (x)(Sx⟶Px), the statement function is Sx⟶Px.

When the statement function is connected to a quantifier, we call it **bound**. When statement functions are bound by quantifiers, we call the statement function's variables **bound variables**. When the statement functions are not bound by quantifiers, we call the statement function's variables **free variables**. It is worth noting that both simple and compound statements in predicate logic can be bound, as illustrated below:

Statements	Bound	Free
Simple	(x)Sx	Sx
Compound	(x)(Sx⟶Px)	Sx⟶Px

As long as the statement function's variables are bound with unasserted quantifier variables (rather than constants), the rules of implication *cannot* be applied. In order to apply the rules of implication, we first need to remove the quantifier, and then assert constants for each variable. When quantifier variables are asserted as constants, we call this **instantiation**. Once a categorical statement has been instantiated, the first eight rules of implication can be applied as they were in propositional logic (whereas, the ten rules of replacement can be used on categorical statements even if they have not been instantiated). There are two rules of inference for instantiating categorical statements in predicate logic, one for instantiating universal statements, and one for instantiating existential statements.

Universal Instantiation (UI)

(x)Px
―――
Pa

or

(x)(Px⟶Qx)
―――――
Pb⟶Qb

or

$$\frac{(x){\sim}Px}{{\sim}Px}$$

or

$$\frac{(x)(Px \longrightarrow {\sim}Qx)}{Py \longrightarrow {\sim}Qy}$$

Notice that in every example of universal instantiation above, the universal quantifier has been removed. In cases where the quantifier binds compound statement functions, the parentheses around the statement functions are removed. Most importantly, **any constant can be used** to instantiate the universal statement, and the constant being used needs to **replace every quantifier variable** previously bound by the universal quantifier. The reason any constant (including the variables x, y, and z) can be used as an instance of the universal statement is that universal statements refer to "all" or "no" members of a subject class participating in the predicate class. Thus, all constants are available to us.

Existential Instantiation (EI)

$$\frac{(\exists x)Px}{Pa}$$

or

$$\frac{(\exists x)(Px \longrightarrow Qx)}{Pb \longrightarrow Qb}$$

or

$$\frac{(\exists x){\sim}Px}{{\sim}Pc}$$

or

$$\frac{(\exists x)(Px \longrightarrow {\sim}Qx)}{Pd \longrightarrow {\sim}Qd}$$

Notice that in every example of existential instantiation above, the existential quantifier has been removed. In cases where the quantifier binds compound statement functions, the parentheses around the statement functions are removed. Most importantly, **we are restricted in which constant can be used** to instantiate each existential statement (even though the constant being used, similarly needs to **replace every quantifier variable** previously bound by the existential quantifier). The reason our choice of constant is restricted has to do with existential statements referring only to "some" members of a subject class participating in the predicate class (at least one). Since we do not know the exact quantity of the participating class, it would be a mistake to assume any quantifier variables or constants already at use in the statement or argument (including the conclusion). Thus, **only unused constants** are available to us.

In addition to needing to instantiate categorical statements in order to apply the rules of implication, in order to apply the change of quantifier rules (which we will learn about in the next section), we need to be able change an instantiated statement into a categorical statement. When asserted constants are removed, replaced with unasserted quantifier variables, and a quantifier is added to the left of the symbolization, we call this **generalization**. Once instantiated statements have been generalized, the change of quantifier rules can be applied. As with instantiation, there are two rules of inference for generalizing categorical statements in predicate logic, one for universal statements, and one for existential statements.

Universal Generalization (UG)

$$\frac{Px}{(x)Px}$$

or

$$\frac{Py \longrightarrow Qy}{(x)(Px \longrightarrow Qx)}$$

or

~Pz
―――――
(x)~Px

or

Px⟶~Qx
―――――
(x)(Px⟶~Qx)

Notice that in every example of universal generalization above, a universal quantifier has been added to the left side of the statement function. In cases of compound statement functions, parentheses have been added around the statement function to show that they are bound to the quantifier. Most importantly, **only instantiated statements with quantifier variables can be generalized** (unlike with universal instantiation), and the quantifier variable 'x' needs to **replace every instantiated variable** previously free in the statement function. The reason we can only generalize from quantifier variables (x, y, and z) is that we cannot assume a universal category of "all" or "none" from a constant, since constants identify only singular subjects. Thus, it would be a mistake to move from a singular assertion to a universal category.

Existential Generalization (EG)

Pa
―――――
(∃x)Px

or

Pb⟶Qb
―――――
(∃x)(Px⟶Qx)

or

~Px
―――――
(∃x)~Px

or

Py⟶~Qy
―――――
(∃x)(Px⟶~Qx)

Notice that in every example of existential generalization above, an existential quantifier has been added to the left side of the statement function. In cases of compound statement functions, parentheses have been added around the statement function to show that they are bound to the quantifier. Most importantly, **any instantiated constant or variable can be generalized**, and the quantifier variable 'x' need only **replace one or more instantiated variable** previously free in the statement function (unlike with universal generalization). The reason we can generalize from any constant or variable is that we know that existential statements require a minimum of one participating class. Thus, we can move from any singular assertion to an existential category.

4.3 Change of Quantifier Rules

We saw in the previous chapter that our first eight rules of inference were powerful tools, but insufficient to derive all conclusions in propositional logic. We therefore added to our system of derivation, rules of replacement. Similarly for predicate logic, using the rules of inference, and our newly introduced generalization and instantiation rules, allow us to derive many conclusions, but not all. So now we need to add what we can call **change of quantifier** rules to use in predicate logic. These rules allow us to move the location of a negation that precedes a quantifier or it's statement function, as well as switch between quantifiers. For instance, using only the methods introduced so far, we have no way of instantiating the following negated universal quantifier: ~(x)Fx

The change of quantifier rules are illustrated below along with English translations that might help illustrate their equivalence:

Quantifier Negation (QN)

Categorical Equivalence	Example of Statement Equivalence
(x)Fx ::~(∃x)~Fx	Everyone is funny :: It is not the case that someone is not funny
~(x)Fx :: (∃x)~Fx	It is not the case that everyone is funny :: Someone is not funny
(∃x)Fx ::~(x)~Fx	Someone is funny :: It is not the case that everyone is not funny
~(∃x)Fx :: (x)~Fx	It is not the case that someone is funny :: Everyone is not funny

Notice that in each instance of quantifier negation, the quantifier on the left has been changed, and both the quantifier and bound statement function on the right have been negated.

As with rules of replacement in propositional logic, change of quantifier rules can be used on part of a line, or a whole line, as long as the rule is applied to all bound statement functions of the quantifier being changed.

To see how the change of quantifier rules work, see the example below:

1.∼(∃x)(Px &∼Qx)	
2.∼(x)(∼Rx∨Qx)	/(∃x)∼Px
3. (x)∼(Px &∼Qx)	1 (QN)
4. (∃x)∼(∼Rx∨Qx)	2 (QN)
5.∼(∼Ra∨Qa)	4 (EI)
6.∼(Pa &∼Qa)	3 (UI)
7.∼∼Ra &∼Qa	5 (DM)
8.∼Pa∨ ∼∼Qa	6 (DM)
9.∼Pa∨Qa	8 (DN)
10.∼Qa	7 (Simp)
11. ∼Pa	9, 10 (DS)
12. (∃x)∼Px	11 (EG)

Notice how the quantifier negation rule is applied to both premises, followed by the resulting lines both being instantiated. However, you will notice that the existential statement on line 4 is instantiated first since we are restricted in our choice of constant. Once those constants have been asserted, we can then repeat them when we instantiate the universal statement on line 3, since there are no restrictions on universal instantiation. Once all categorical statements have been instantiated, we apply the rules of inference accordingly to remove all parentheses and derive the statement on line 11 which is then generalized to derive the desired conclusion.

To see how the change of quantifier rules can work on only part of some line, see the example below:

1. (∃x)Px⟶∼(∃x)Qx	
2. (x)∼Qx⟶(x)∼Rx	/(∃x)Px⟶∼(∃x)Rx
3. (∃x)Px⟶(x)∼Qx	1 (QN)
4. (∃x)Px⟶(x)∼Rx	2, 3 (HS)
5. (∃x)Px⟶∼(∃x)Rx	4 (QN)

Notice how the quantifier negation rule is only applied to the consequent of line 1 and then later on, the consequent of line 4, in order to arrive at places where the rules of inference can be applied.

4.4 Conditional and Indirect Proof for Predicate Logic

Conditional and indirect proof work very similarly in predicate logic as they do in propositional logic. We saw in propositional logic that without conditional and indirect proof, many conclusions would be impossible or, at the very least, quite difficult to derive without them. The same goes for conclusions that we want to derive in predicate logic. Our strategies will thus be quite similar, for conclusions which are conditional statements, conditional proof will often be appropriate, whereas conclusions that are particular statements are immediate candidates for indirect proof.

To see how to use conditional proof to derive a conclusion with two bound variables, see the example below:

1. (x)(Tx⟶Ux)	/ (∃x)Tx⟶(∃x)Ux
│ 2. (∃x)Tx	(ACP)
│ 3. Ta	2 (EI)
│ 4. Ta⟶Ua	1 (UI)
│ 5. Ua	3, 4 (MP)
│ 6. (∃x)Ux	5 (EG)
7. (∃x)Tx⟶(∃x)Ux	2-6 (CP)

Notice here that we assumed the antecedent of the desired conclusion on line 2, then we derived the consequent that we needed in the subderivation on line 6. Finally, we discharged the entire conditional statement to arrive at the desired conditional statement.

To see how to use conditional proof to derive a conclusion that has one universally bound quantifier, see the example below:

1. $(x)[(Px \lor Qx) \longrightarrow Rx]$	/ $(x)(Px \longrightarrow Rx)$
	2. Px	(ACP)
	3. $Px \lor Qx$	2 (Add)
	4. $(Px \lor Qx) \longrightarrow Rx$	1 (UI)
	5. Rx	3, 4 (MP)
6. $Px \longrightarrow Rx$	2-5 (CP)
7. $(x)(Px \longrightarrow Rx)$	6 (UG)

Notice here that since we cannot assume a universalized Px, because it is bound in the conclusion, we must first derive the conditional of the statement function and then generalize it to arrive at the desired conclusion.

This leads us to an important **restriction** that we need to place on **universal generalization (UG)**. In a subderivation, we cannot universally generalize on an unbound or free variable that is our assumption. To see this, look at the following proof:

1. $(x)[(Px \lor Qx) \longrightarrow Rx]$	/ $(x)(Px \longrightarrow Rx)$
	2. Px	(ACP)
	3. $(x)Px$	2 (UG) **INVALID**

To understand this restriction, we can first review our previous restriction on universal generalization (UG) in that we cannot assume that "all" of something is the case simply because "one" is the case. Additionally, if we were to violate this general restriction in a subderivation, we would violate the purpose of derivations (to apply valid rules of inference) by deriving a false conclusion from true premises. Thus, this restriction prevents us from making derivations in an invalid argument.

Taking the above example, let's let 'Px' stand for "x is a pony" and let's let 'Rx' be "x is a rattlesnake". If we allowed particular statements to be universally generalized within the subderivation of the argument above, then our statement (x)(Px⟶Rx) would mean that "for all x, if x is a pony, then x is a rattlesnake". Additionally, if we did not adhere to this restriction within the subderivation, we would be able to derive the following inference from the given premises: "all ponies are rattlesnakes" (which we obviously should not want, if our system of derivation is to be sound and complete)[3].

To see how to use indirect proof, see the example below:

1. $(\exists x)Rx \lor (\exists x)Sx$	
2. $(x)(Rx \longrightarrow Sx)$	/ $(\exists x)Sx$
	3. $\sim(\exists x)Sx$	(AIP)
	4. $(\exists x)Rx$	1, 3 (DS)
	5. Rc	4 (EI)
	6. $Rc \longrightarrow Sc$	2 (UI)
	7. Sc	5, 6 (MP)
	8. $(x)\sim Sx$	3 (QN)
	9. $\sim Sc$	8 (UI)
	10. $Sc \mathbin{\&} \sim Sc$	7, 9 (Conj)
11. $\sim\sim(\exists x)Sx$	3-10 (IP)
12. $(\exists x)Sx$	11 (DN)

[3]For more on this see Grayling (2001)

An Overview of Rules for Predicate Logic

<table>
<tr><td>Universal Instantiation (UI)
$\frac{(x)Fx}{Fy}$ $\frac{(x)Fx}{Fa}$</td><td>Universal Generalization (UG)
$\frac{Fy}{(x)Fx}$ invalid: $\frac{Fa}{(x)Fx}$</td></tr>
<tr><td>Existential Instantiation (EI)
$\frac{(\exists x)Fx}{Fa}$ invalid: $\frac{(\exists x)Fx}{Fy}$</td><td>Existential Generalization (EG)
$\frac{Fa}{(\exists x)Fx}$ $\frac{Fy}{(\exists x)Fx}$</td></tr>
<tr><td colspan="2">The first eight rules of implication can only be applied to instantiated categorical statements or to whole lines where the main operator is outside the scope of the quantifier.</td></tr>
<tr><td colspan="2">The ten rules of replacement can be applied to all categorical statements.</td></tr>
<tr><td colspan="2">Quantifier Negation (QN)
$(x)Fx :: \sim(\exists x)\sim Fx$ $(\exists x)Fx :: \sim(x)\sim Fx$
$\sim(x)Fx :: (\exists x)\sim Fx$ $\sim(\exists x)Fx :: (x)\sim Fx$</td></tr>
</table>

Bibliography

[1] Baronett, Stan. *Logic*. 3rd edition, Oxford University Press, 2015.

[2] Grayling, A.C. *An Introduction to Philosophical Logic*. 3rd edition, Wiley-Blackwell, 2001.

[3] Hurley, Patrick and Watson, Lori. *A Concise Introduction to Logic*. 13th edition, Cengage, 2018.

[4] Nolt, John. *Schaum's Outline of Logic*. Dennis Rohatyn and Achille Varzi, 2nd edition, McGraw-Hill, 2011.

[5] van Fraassen, Bas. *The Scientific Image*. Clarendon Press, 1980.