Theodore G. Faticoni

Combinatorics

An Introduction

WILEY

Contents

Combinatorics
An Introduction

Theodore G. Faticoni
Department of Mathematics
Fordham University
Bronx, NY

WILEY
A JOHN WILEY & SONS, INC., PUBLICATION

Cover Design: John Wiley & Sons, Inc.
Cover Illustration: courtesy of Theodore G. Faticoni

Published by John Wiley & Sons, Inc., Hoboken, New Jersey.
Published simultaneously in Canada.

For general information on our other products and services please contact our Customer Care Department within the United States at (800) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print, however, may not be available in electronic formats. For more information about Wiley products, visit our web site at www.wiley.com.

Library of Congress Cataloging-in-Publication Data:

Faticoni, Theodore G. (Theodore Gerard), 1954–
Combinatorics : an introduction / Theodore G. Faticoni, Department of Mathematics, Fordham
University, Bronx, NY.
pages cm
Includes bibliographical references and index.
ISBN 978-1-118-40436-2 (hardback)
1. Combinatorial analysis. I. Title.
QA164.F38 2012
511’.6—dc23

2012025751

To our mother, Margaret Faticoni.
My sisters and I are one for our Mom ‘s efforts.

Preface

As I read the current textbooks on finite or linear mathematics, I am struck by the superficial way that counting problems or combi- natorics are handled. Counting is treated as a methodical or me- chanical thing. The student is asked to memorize a few important but unenlightening algorithms that will always teil us the number of ways that someone can choose and arrange her outfits for the week.

Furthermore, the examples that are given use so much of reality that the Student has more to learn about electronic components, failed tests, and card games than they do about counting in mathematics. Whatever happened to problems that emphasized their mathematical content and left a knowledge of Science and gaming to other departments? I understand that, to some, mathematics is best when it is used in applications. But why are we giving up on teaching mathematical content in favor of these other subjects?

Moreover, the why of it all, the justification, the beauty of proof has left these courses entirely. There is no explanation as to how the fundamental formulas are derived, and there is no rationalization as to how certain formulas are formed. The exercises that are given in modern texts are just slight variations on the examples worked out in the chapter. And in my opinion, the chapter examples are mostly uninspiring.

This book is aimed at College students, teaching assistants, ad- junct instructors, or anyone who wants to learn a little more el- ementary combinatorics than the usual text contains. This book might also-be used as a Supplement to the existing text for a finite mathematics course or to Supplement a discrete mathematics course, which several curriculums require.

The purpose of this book is to give a treatment of counting combinatorics that allows for some discussion beyond what is seen in today's texts. We will discuss and justify our formulas at every turn. Our examples will include, after the most elementary of appli- cations, some ideas that do not occur in other texts on the market at this time. The applications never get beyond the use of Venn diagrams, the inclusion/exclusion formula, the multiplication principal, permutations, and combinations. But their uses are clever and at times inspiring.

For example, we do some poker hand problems that are not seen in modern texts, we count the number of bracelets that can be made with $n > 1$ different colored beads, and we count the number of derangements of $\{1, \ldots, n\}$. We do this without any more than the elementary tools for counting. We then consider some probability problems by doing some elementary counting. But we show some very surprising, mathematically precise consequences of a trained approach to the subject.

A second theme within this book is that the case-by-case method for solving problems is emphasized. Of course we use a formula when needed, but when it comes time to derive a formula, we have decided to consistently give the case-by-case approach to the Problem. In this way we are asking the Student/reader to think mathematically and in exactly the same way from problem to problem throughout the book. Perhaps this is what the students will take with them when they leave the course. They will misremember the applications for the permutation formula, but they might remember how to break a problem into pieces in order to solve it.

The book is a series of short chapters that cover no more than one topic each. We cover such topics as logic and paradoxes, sets and set notation, power sets and their cardinality, Venn diagrams, the multiplication principal, permutations, combinations, problems combining the multiplication principal, problems combining permutations and combinations, problems involving the complement rule, *at least*, and *at most.* We cover derangements, elementary probability, conditional probability, independent probability, and Bayes' Theorem. We close with a discussion of two dimensional geometric simplex algorithm problems, showing that the traditional geometric method breaks down in the case where the variables take on only integer values. In other words, the method breaks down in every example done in the modern finite mathematics texts.

There are plenty of worked examples, as I want to do the work for the reader, and there is a short list of homework exercises. The examples given can also be used by an instructor or a teaching assis- tant to gain a higher level of understanding of the subject than the current texts offer, thus providing the instructor with an overview of the subject that the Student does not possess. This can aid the classroom Situation since, as I believe, we do a better job of teaching when we teach from a higher point of view in the delivered subject. The instructor then has a Professional confidence that (s)he can solve any problem that comes up in dass.

The fact that the book is salted with explanations as to why cer- tain formulas exist helps the Student and the instructor understand what they are doing. This is different from the rote memorization that many texts on this subject require. In this book the justifica- tion for the formulas is also there.

With this approach to the subject and to my readers, I believe I have found a gradual, understandable path that will bring a College student to a discussion of a subject on combinatorics and probability that is more advanced than any of the topics covered in the current texts on finite mathematics.

Theodore G. Faticoni
Department of Mathematics
Fordham University
Bronx, New York 10458
faticoni@fordham.edu

Chapter 1

Logic

There are several kinds of logic in mathematics. The one based in the construction of Truth tables is called *formal logic*. This is the logic used in Computer science to design and construct the guts of your Computer. And then there is Aristotle's logic. This is the logic used to make arguments in court or when arguing informally with another person. This is the logic used to prove that something is, or to prove that something is not. This is the logic used to examine combinations of any of the mathematical ideas encountered in this text. While we will examine formal logic and the logic of sets and functions, we will be most interested in Aristotle's *logic of the argument* in this chapter and throughout the rest of the text.

Oh, and there will be no need for a calculator in this book. I have made an effort to emphasize the important mathematical content in this book, not the superfluous, tedious practice of arithmetic. Arithmetic is important when you work with money, but in more challenging mathematical problems it only gets in the way. So cradle your electronic toy if you need to, but there will be almost no use for it as we do our counting.

1.1 Formal Logic

Formal logic is just a series of tables describing how the words *and, or, not* are defined. There is nothing illuminating with this approach, but it does match the operations of the inner workings of your Computer. We will minimally justify the tables used here. We will just write them down and show how they agree with your use of the words in your language.

These tables define logic. Not just in English, the language that this book is being written in, but they describe logic in *every* language on earth. If you are reading a Mandarin Chinese translation of this book, then the logic presented here will still be the logic of your language. It is also the binary language in which the Software in your Computer is written. Take time to savor that thought. Logic as it is applied to languages and Computers is universal. Logic is thus common to all forms of communication, analogue or digital.

To begin with we need to know what the logical operations are and what they operate on. *Logic operates on statements*, and ordinarily we will use the letters *P*, *Q*, and *R* to denote the statements that we we are working on. These statements can take on the *logical states T* (for True) and *F* (for False).

You already have an intuitive understanding' of what it means for a statement to be True or False. You know that *The sky is blue* is True on earth, and you know that *You and I are human* is a True statement. *You have five dollars* might be True right now, but it might be False come late Friday evening. Of course *R is raining* is a False statement on a sunny day over my home, but it might be a True statement for you where you live. So let us assume that we know what *T* and *F* mean in this context.

The first logical operation that we will investigate is the Operation *not*. The *not* operation takes a statement *P* and changes or negates its logical states. It changes *T* to *F* and *F* to *T*. Its *Truth table*, the table that lists the logical states of the *not* operation, follows.

P	not *P*
T	*F*

F	*T*

This is just a tabular way of defining what *not* is. Notice that according to the table, if *P* is *T* then not *P* is *F*, and if *P* is *F* then not *P* is *T*. As we said, *not* changes a statement's logical state to the complementary logical state.

EXAMPLE 1.1.1 1. If *P* is the statement *The sky is blue on earth*, then not *P* is the statement *The sky is not blue on earth*. We have negated *P* and changed its logical state from *T* to *F*.

2. If *P* is $1 + 2 = 3$ then not *P* is the statement $1 + 2 \neq 3$. Again the logical state of *P* has been changed by an application of *not* from *T* to *F*.

Because of the nature of the word *not*, two consecutive applications of the operation *not* to *P* will leave the logical states of *P* unchanged. For lingual reasons we let not not *P* = not(not *P*). In tabular form the compound operation *not not* is written as follows.

P	not *P*	not (not *P*)
T	*F*	*T*
F	*T*	*F*

Notice that if *P* is *T* then not *P* is *F*, and then not(not *P*) is *T*, giving not(not *P*) the logical states of *P*. You know this as a *double negative* from your English dass.

EXAMPLE 1.1.2 1. If *P* is *The sky is blue on earth*, then the double negative not (not *P*) is the awkward sentence *It is False that the sky is not blue on earth*. Your language skills compel you to avoid the double negative and just write *The sky is blue on earth*.

2. Suppose *P* is *I think this is wrong*. Then not *P* is *I think this is not wrong*, and not(not *P*) is the very awkward *I don't think that this is not wrong*. You would be advised by your language teacher to avoid the double negative and just say *I think this is wrong*. The statements *P* and not (not *P*) are

written with different words, but logically they express the same meaning.

Thus, by applying the logic of the operator *not* to a lingual double negative, we can avoid the double *not*.

Throughout this discussion, suppose that we are given statements P, Q. Several logical operations allow us to compare the logical states of P, Q by combining them.

For instance, we can combine statements P, Q using the *and* operation. This is the *and* that you use all of the time when you write. When applied to P, Q the *and* operation yields the statement "P and Q". This is just the compound statement formed by combing P, Q with the conjunction *and* from English.

EXAMPLE 1.1.3 1. If P is *The sky is blue on Earth* and if Q is *You are a man* then "P and Q" is the statement *The sky is blue on Earth and you are a man*.

2. If P is *This is wrong* and if Q is *These are red* then "P and Q" is *This is wrong and these are red*.

The logical states of P and Q are closely related to the way that the word *and* behaves in language. Thus the logical state of P and Q is T (True) exactly when both P and Q are T. In every other instance, "P and Q" is F (False). Put another way, if one or more of the logical states of P, Q are F (False) then the statement "P and Q" is a Falsehood, its logical value is F.

In the form of a Truth table the *and* operation is diagrammed as follows:

P	Q	P and Q
T	T	T
T	F	F
F	T	F
F	F	F

The first row states that if both P, Q have logical state T then the conjunction "P and Q" also has logical state T.

Once we know that the right hand entry of the first line in the table is T then the rest of the rows follow as F.

EXAMPLE 1.1.4 1. If P is *I am a human being* and if Q is *I am sitting in my chair* then "P and Q" is T exactly when *I am a human being* is T and *I am sitting in my chair* is T. Any other combination of T's and F's for P, Q will produce a logical state F for "P and Q".

2. If P is *The sky is red over me* and if Q is *The ground is dry beneath me* then the logical value of "P and Q" is F if we are on Earthsince the sky is not red there. If we are on Mars then the logical value of "P and Q" is T because the sky is red and the ground is dry on Mars.

Another way to combine statements is through the use of the conjunction *or*. The use of *or* in logic is denoted by the operation *or*. Thus, statements P, Q are combined to form the conjunctive statement "P or Q", which is read just like the *or* statements that you read and write.

The compound statement "P or Q" has logical state T exactly when one or more of the statements has logical state T. But it might be easier to remember how *or* behaves with False statements. When the logical states of both P and Q are F then "P or Q" has logical state F, and this is the only case in which the logical state of "P or Q" is F.

We will always use the *inclusive or* here so that the statement "P or Q" includes the case where both P, Q have logical state T. That is, we we read "P or Q" as *P, Q, or both P and Q*.

EXAMPLE 1.1.5 1. If P is *The river is wide* and if Q is *The water is cold* then "P or Q" is read as *The river is wide or the water is cold*. Since "P or Q" is T when either P, Q has logical state T, the compound statement *The river is wide or the water is cold* has logical state T if the river is wide.

2. *The river is wide or the water is cold* is T if we are talking about the Missouri River and its waters are cold. *The river is wide or the water is cold* is T if we are talking about

the Missouri River and the water we are talking about is in my coffee.

3. Let P be the statement *All is nothing* and let Q be the arithmetical statement $1 + 1 = 3$. Both P and Q have logical state F, so that "P or Q" has logical state F. Since both P, Q have logical state F then "P or Q" has logical state F.

The next logical operations, called *DeMorgan's laws*, show us how the logical operations *and, or, not* combine with each other. Simply put, *DeMorgan's laws* are lingual ways of simplifying a sentence that uses *and* , *or*, and *not* is a more complex manner.

Given statements P, Q then *DeMorgan's laws* are written as

$$\text{not}(P \text{ or } Q) = (\text{not } P) \text{ and } (\text{not } Q)$$
$$\text{not}(P \text{ and } Q) = (\text{not } P) \text{ or } (\text{not } Q).$$

Notice that in our use of DeMorgan's Law, the distribution of the *not* operator changes *or* to *and*, or it changes *and* to *or*. Compare this to the following lingual examples of uses of DeMorgan's laws. When read properly, you will see that the symbolism we use here is the same as our use of *and, or, not* above.

We will use parentheses to emphasize a statement's meaning, so that there is no confusion as to what word modifies what phrase.

EXAMPLE 1.1.6 1. The statement

(The river is not wide) or (the water is not cold)

is equivalent to the statement

It is not True that (The river is wide and the water is cold).

Complex to be sure, but that is the purpose behind DeMorgan's laws. It will take a complicated statement and make it easier to read.

2. The statement

(This is not a king) and (this is not a queen), is equivalent to the statement

This is not (a king or a queen).

3. The statement

This box does not contain (a red and a yellow crayon),

is equivalent to

(This box does not contain a red crayon) or
(it does not contain a yellow crayon).

EXAMPLE 1.1.7 1. Let P be the statement that *This is a king* and let Q be the statement that *This is a queen*. The statement "not (P or Q)" is also written as

It is False that (this is a king or a queen),

while "(not P) and (not Q)" is written as

(This is not a king) and (this is not a queen).

Which do you prefer? Logically they both mean the same thing.

2. Let P be the statement that *This box contains a red crayon* and let Q be *This box contains a yellow crayon*. Then "not (P and Q)" is written as

It is False that (this box contains a red and yellow crayon),

while its equivalent formulation "(not P) or (not Q)" under DeMorgan's laws is

(This box does not contain a red crayon) or
(this box does not contain a yellow crayon).

1.2 Basic Logical Strategies

We will make exclusive use of logical arguments due to Aristotle some 500 years B.C. They are the basis for every intelligent conversation and every legal argument made since.

The first logical observation is that one statement always has a logical state of F.

The statement “P and (not P)” is a universal Falsehood.

No matter what the logical state of P is, “P and (not P)” is a Falsehood.

To see this, notice that because *not* changes logical states, at any time either P or not P is F. Thus the *and* statement “P and (not P)” has logical state F. The Truth table for “P and (not P)” is then given as follows:

P	not P	P and (not P)
T	F	F
F	T	F

Observe that the right-hand column of the table is made up of F’s. Thus, the statement “P and (not P)” is a Falsehood.

EXAMPLE 1.2.1 1. Let P be *The sky is blue*. Then (*the sky is blue) and (the sky is not blue)* is a Falsehood.

2. Let P be *This statement is True*. Then “P and (not P)” is the statement *This statement is True and this statement is not True*, and this is a Falsehood.

3. Let P be *There is a mountain*. Then “P and (not P)” is *(There is a mountain) and (there is no mountain)*, which is a Falsehood. So is *First there is a mountain, then there is no mountain, then there is*.

We continue our discussion of logical arguments. Given statements P, Q, the statement “P implies Q” is called an *implication*, and it is symbolically written as

$$P \Rightarrow Q.$$

The statement P is called the *premise of the implication* and Q is called its *conclusion*.

The logical states of $P \Rightarrow Q$ are determined by one line of explanation.

If your argument is correct then Truth leads to Truth.

In other words, if your argument is T and if your premise P is T then your conclusion Q is T. Every other logical state of $P \Rightarrow Q$ follows from this boxed statement.

Note that line one of the following Truth table for "$P \Rightarrow Q$" is logically equivalent to the boxed statement above.

P	Q	$P \Rightarrow Q$
T	T	T
T	F	F
F	T	T
F	F	T

Let us fill in the remaining Truth values for this table. Let P and Q be statements and consider "$P \Rightarrow Q$". We will show how a few simple Truths about argument discovered by Aristotle can be used to fill in the Truth table for the *implication*.

EXAMPLE 1.2.2 We will continually refer to the Truth table for "$P \Rightarrow Q$".

1. Because Truth implies Truth when the argument is correct,

If your argument is correct (T), and if P is T then Q is T.

This is why line 1 is $\begin{array}{cc|c} P & Q & P \Rightarrow Q \\ \hline T & T & T \end{array}$.

2. Since Truth implies Truth when the argument is correct,

Your argument is False if P is T and Q is F.

This is why line 2 of the Truth table is $\begin{array}{cc|c} P & Q & P \Rightarrow Q \\ \hline T & F & F \end{array}$.

3. Since any argument begun with a False premise is correct, we can write

---

Your argument is T if P is F.

This is why lines 3 and 4 of the Truth table ar

P	Q	$P \Rightarrow Q$
T	F	F

. The column under Q is the list of all possible logical states for Q in the Truth table for "$P \Rightarrow Q$".

4. Since a False premise leads to either a True or False conclusion,

---

Your conclusion is ambiguous if P is F.

This is why lines 3 and 4 of the Truth table are

P	Q	$P \Rightarrow Q$
T	F	F

. The column under Q completely describes an ambiguous conclusion Q. The T's under "$P \Rightarrow Q$" result from the part 3.

Let us put this implication to work in some elementary arguments.

EXAMPLE 1.2.3 1. Here is a Greek classic. We will use Example 1.2.2(1). Begin with P : *Socrates is a man*. The conclusion will be Q : *Socrates is mortal*. The implication $P \Rightarrow Q$ is *If Socrates is a man then Socrates is mortal*. Since the implication $P \Rightarrow Q$ is correct, and since the Truth of the premise P implies the Truth of the conclusion Q, Socrates is mortal.

2. The premise is P : *I stand on dry land on earth*, and the conclusion is Q : *The sky above me is blue*. The implication is *If I stand on dry land on Earth then the sky above me is blue* is True. Since P is True, and since Truth leads to Truth, Q is True.

3. The premise is P : *Digital technology is like pockets*, and the conclusion is Q : *We have had digital technology for hundreds of years*. The implication is "$P \Rightarrow Q$" *We have had pockets for hundreds of years*. Let us assume that the premise P is True. Since Q is Falsehood, the implication "$P \Rightarrow Q$" has logical state F. But if we assume that the premise P

is False, then Q is still False, but the implication "$P \Rightarrow Q$" is True.

4. Under what conditions will P in part 3 lead us to a True conclusion Q? Have fun with this one.

1.3 The Direct Argument

This formal manipulation of statements is not exactly what we are interested in for this chapter. It is good to know that an argument has logical state T or F, but it is better to know how we can use the implication to correctly deduce a conclusion.

The first line T, T, T of the Truth table for $P \Rightarrow Q$ can be restated as *If our argument is correct then Truth leads to Truth*, or in other words, *If the premise is True and if the argument is correct then the conclusion is True*. This form of argument is called the *direct argument*. It is not new to you since you unconsciously use direct arguments in your everyday life.

EXAMPLE 1.3.1 1. The premise is P : *The sky is not blue* and the conclusion is Q : *We are not on earth*. A correct argument is

If the sky is not blue then we are not on earth.

Conclude that the conclusion Q is True.

2. Something more mathematical begins like this. The premise is $P : 1 + 1 = 2$. Argue correctly as follows:

$1 + 1 = 2$

If we add 1 to both sides of $1 + 1 = 2$ then $1 + 1 + 1 = 2 + 1$.

If $2 + 1 = 3$ then $1 + 1 + 1 = 3$.

The conclusion $Q : 1 + 1 + 1 = 3$ is then True.

A chain-like form of argument shows us the structure inherent in longer arguments called *transitive property*. These longer arguments are what people make when they logically move from one idea to the next. Basically, the *transitive property of implications* is a way to leap from two or more implications to one implication. Hence

If $P \Rightarrow Q$ and if $Q \Rightarrow R$ then $P \Rightarrow R$.

A series of implications and the transitive property provide us with a method for arguing efficiently with many implications. This series of implications is called the *transitive argument*.

Assume the Truth of the premise P.
Show that $P \Rightarrow Q$ is True
Show that $Q \Rightarrow R$ is True
Conclude the Truth of R.

To justify that this column forms an argument that we can use to deduce R from P, we will argue lingually.

Proof: Assume the Truth of P. If $P \Rightarrow Q$ is True then by the Direct Argument Q is True. If $Q \Rightarrow R$ is True then by the Direct Argument we conclude the Truth of R. Therefore, our transitive argument concludes the Truth of R from the Truth of P.

Let us review what we just argued in terms of True statements. We begin with a True statement P. The assumption is that $P \Rightarrow Q$ and $Q \Rightarrow R$ are True, which allows us to make a correct transitive argument

$$P \Rightarrow Q \text{ and } Q \Rightarrow R \text{ implies } P \Rightarrow R.$$

From the Truth of P and the Truth of $P \Rightarrow R$ we use the Direct Argument to conclude the Truth of R.

In a later section we will argue as we did above and in greater detail, thus producing three more argument forms.

EXAMPLE 1.3.2 This example shows how the above discussion can be applied to longer arguments.

a) The premise is P: $10 < 2^{10}$.

b) $P \Rightarrow Q$: Because $10 < 2^{10} = 1024$ then $11 < 2^{10}$.

c) $Q \Rightarrow R$: Because $11 < 2^{10}$ then $11 < 2 \cdot 2^{10} = 2^{11}$.

d) Conclude R: $11 < 2^{11}$.

Using this iterated form of argument people form longer and more complicated arguments, which allows them to perform more complicated intellectual tasks. These tasks could be just a way of adding numbers, or it could be the design of your computer's Software, or it could be that the arguments take the arguer to intellectual places that no one had conceived before. The lesson to learn here is that, while the tabular thinking of logic is good for some tasks, there will always come a time in problem solving when we must use argument and a more enlightened form of thinking if we are to make progress on hard problems.

REMARK 1.3.3 When your Computer operates it is working its way through a very long and tedious argument based on the very simple *binary logic* introduced in this section. The steps in the computer's argument are mechanical, a form of arithmetic completed by a machine. The men and women who designed this Computer had to think through the *binary logic* during the implementation phase of the Software.

However, for the men and women who put the larger internal logical parts of the Computer together in the design phase, the problems encountered could not be solved with a simple manipulation of *binary logic*. They had to think creatively through the problems presented to them by the design phase. These solutions would often include a leap of the imagination that could not be anticipated when the design for the Computer was initially proposed. The logical problems yet to come will require those leaps of the imagination before we can solve our problems.

1.4 More Argument Forms

Converse Statements

The implication $P \Rightarrow Q$ comes with what is called its *converse*.

The converse of $P \Rightarrow Q$ is $Q \Rightarrow P$.

Let us write down the Truth table for $Q \Rightarrow P$ and compare it to $P \Rightarrow Q$.

P	Q	$P \Rightarrow Q$
T	T	T
T	F	F
F	T	T
F	F	T

P	Q	$Q \Rightarrow P$
T	T	T
T	F	T
F	T	F
F	F	T

As you can see, the implication and its converse do not have the same Truth table. The logical state of the implication $P \Rightarrow Q$ in the third row is T, while the logical state of the implication $Q \Rightarrow P$ in the third row is F. Thus the converse implication $Q \Rightarrow P$ can have logical state F even when $P \Rightarrow Q$ has logical state T. For this reason, the converse cannot be used as a True statement even when the original implication is True. Hence *all* are forewarned to avoid the classic error of using the converse of an implication to advance an argument.

EXAMPLE 1.4.1 These examples show that we cannot interchange the implication with its converse. They will have different logical states.

1. Let P be the T statement *The sky is blue*, and let Q be *The world is flat*. Then "$P \Rightarrow Q$" is F.

The converse of "$P \Rightarrow Q$" is the statement "$Q \Rightarrow P$:" *If the world is flat then the sky is blue*. Since its premise Q is F, "$Q \Rightarrow P$" is T. Thus the implication is False while the converse is True, and we cannot exchange them in arguments or conversation.

2. The implication is *If today is Monday then my schedule is clear* and its converse is *If my schedule is clear then today is Monday*. The implication may be True, but the converse is False since my schedule is clear on Sunday.

Contrapositive Statements

Suppose that we consider the implication $P \Rightarrow Q$, assuming that it is T. If Q is F then the Truth table for $P \Rightarrow Q$ shows us that P is also F. Thus, a False premise Q implies a False conclusion P. This is an important implication known as the *contrapositive*.

$$\text{not } Q \Rightarrow \text{not } P.$$

When one writes out the Truth table for the implication and its contrapositive, a curious things occurs. This Truth table reveals that the two arguments have identical Truth tables.

P Q	$P \Rightarrow Q$	not $Q \Rightarrow$ not P
T T	T	T
T F	F	F
F T	T	T
F F	T	T

Notice that the rightmost two columns are identical lists of T's and F's. This is completely different from what we found with the converse. The table shows that

---

The implication and its converse are logically equivalent. One can be substituted for the other without loss of Truth.

In other words, the statements "$P \Rightarrow Q$" and "not $Q \Rightarrow$ not P" are both True for the same logical values of P and Q.

EXAMPLE 1.4.2 1. The implication *If the sky is not blue then this is not earth* has as contrapositive *If this is earth then the sky is blue*. The implication and its contrapositive are making the same logical statement about the sky.

2. The implication *If my GPS is working then I am not lost* has contrapositive *If I am lost then my GPS is not working*. Notice that both the implication and its contrapositive are making the same logical statement, assuming I always use my GPS.

3. The implication *If my spell-check program is running then I do not misspell all the time* has contrapositive *If I misspell all the time then my spell-check program is not running*. Notice that both the implication and its contrapositive make the same logical statement about a man who cannot spell without technological help.

Counterexamples

The next form of argument does not use T's and F's. It is strictly lingual.

Let P be a statement. A *counterexample to P* is an example that is in logical conflict with the content of P. The existence of a counterexample to P proves that P is False.

The idea behind the proof by counterexample is this. If I claim that P : *All colors are white* is True then you can disprove my claim by *producing some color that is not white*. One non-white color will do. I choose red. With the existence of the color red you have refuted my claim. You have proved that *All colors are white* is a Falsehood.

In the very same manner, we can disprove any statement that asserts that all of the X's in the world are short Y's. All we need do is find a counterexample X that is not a short Y.

The *proof by counterexample* can be summed up as follows:

> The statement *All X 's have property Y* is **disproved** by a counterexample of an X that does not have property Y.

These proofs by counterexample all proceed in the same way. Producing just one X that *does not have* quality Y is