
Discrete Maths: Exercises & Solutions

Propositional Logic

Logic

The rules of logic specify the meaning of mathematical statements. Logic is the basis of all mathematical reasoning, and of all automated reasoning. It has practical applications to the design of computing machines, to the specification of systems, to artificial intelligence, to computer programming, to programming languages, as well as, to many other fields of study.

1.1 Propositional Logic:

Introduction: The rules of logic are used to distinguish between valid and invalid mathematical arguments. Logic has numerous applications in e.g. the construction of computer programs, the verification of the correctness of programs etc.

Propositions: A **proposition** is a declarative sentence (that is, a sentence that declares a fact) that is either true or false, but not both.

EXAMPLE 1 All the following declarative sentences are propositions.

1. Washington, D.C., is the capital of the United States of America.
2. Toronto is the capital of Canada.
3. $1 + 1 = 2$.
4. $2 + 2 = 3$.

Propositions 1 and 3 are true, whereas 2 and 4 are false.

Some sentences that are not propositions are given in Example 2.

EXAMPLE 2 Consider the following sentences.

1. What time is it?
2. Read this carefully.
3. $x + 1 = 2$.
4. $x + y = z$.

Sentences 1 and 2 are not propositions because they are not declarative sentences. Sentences 3 and 4 are not propositions because they are neither true nor false. Note that each of sentences 3 and 4 can be turned into a proposition if we assign values to the variables. We will also discuss other ways to turn sentences such as these into propositions in Section 1.4.

We use letters to denote **propositional variables** (or **statement variables**), that is, variables that represent propositions. The conventional letters used for propositional variables are p, q, r, s, \dots . The **truth value** of a proposition is true, denoted by T, if it is a true proposition, and the truth value of a proposition is false, denoted by F, if it is a false proposition.

The area of logic that deals with propositions is called the **propositional logic**. It was first developed systematically by the Greek philosopher Aristotle more than 2300 years ago.

We have methods for producing new propositions from those that we already have. Many mathematical statements are constructed by combining one or more propositions. New propositions, called **compound propositions**, are formed from existing propositions using logical operators.

DEFINITION 1

Let p be a proposition. The *negation of p* , denoted by $\neg p$ (also denoted by \bar{p}), is the statement
“It is not the case that p .”

The proposition $\neg p$ is read “not p .” The truth value of the negation of p , $\neg p$, is the opposite of the truth value of p .

EXAMPLE 3 Find the negation of the proposition

“Michael’s PC runs Linux”

and express this in simple English.

Solution: The negation is

“It is not the case that Michael’s PC runs Linux.”

This negation can be more simply expressed as

“Michael’s PC does not run Linux.”

EXAMPLE 4 Find the negation of the proposition

“Vandana’s smartphone has at least 32GB of memory”

and express this in simple English.

Solution: The negation is

“It is not the case that Vandana’s smartphone has at least 32GB of memory.”

This negation can also be expressed as

“Vandana’s smartphone does not have at least 32GB of memory”

or even more simply as

“Vandana’s smartphone has less than 32GB of memory.”

Table 1 displays the **truth table** for the negation of a proposition p . This table has a row for each of the two possible truth values of a proposition p . Each row shows the truth value of $\neg p$ corresponding to the truth value of p for this row.

TABLE 1 The Truth Table for the Negation of a Proposition.

p	$\neg p$
T	F
F	T

The negation of a proposition can also be considered the result of the operation of the **negation operator** on a proposition. The negation operator constructs a new proposition from a single existing proposition. We will now introduce the logical operators that are used to form new propositions from two or more existing propositions. These logical operators are also called **connectives**.

DEFINITION 2 Let p and q be propositions. The *conjunction* of p and q , denoted by $p \wedge q$, is the proposition “ p and q .” The conjunction $p \wedge q$ is true when both p and q are true and is false otherwise.

Table 2 displays the truth table of $p \wedge q$. This table has a row for each of the four possible combinations of truth values of p and q . The four rows correspond to the pairs of truth values TT, TF, FT, and FF, where the first truth value in the pair is the truth value of p and the second truth value is the truth value of q .

TABLE 2 The Truth Table for the Conjunction of Two Propositions.

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

EXAMPLE 5 Find the conjunction of the propositions p and q where p is the proposition “Rebecca’s PC has more than 16 GB free hard disk space” and q is the proposition “The processor in Rebecca’s PC runs faster than 1 GHz.”

Solution: The conjunction of these propositions, $p \wedge q$, is the proposition “Rebecca’s PC has more than 16 GB free hard disk space, and the processor in Rebecca’s PC runs faster than 1 GHz.” This conjunction can be expressed more simply as “Rebecca’s PC has more than 16 GB

free hard disk space, and its processor runs faster than 1 GHz.” For this conjunction to be true, both conditions given must be true. It is false, when one or both of these conditions are false.

DEFINITION 3 Let p and q be propositions. The *disjunction* of p and q , denoted by $p \vee q$, is the proposition “ p or q .” The disjunction $p \vee q$ is false when both p and q are false and is true otherwise.

Table 3 displays the truth table for $p \vee q$.

TABLE 3 The Truth Table for the Disjunction of Two Propositions.		
p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

The use of the connective *or* in a disjunction corresponds to one of the two ways the word *or* is used in English, namely, as an **inclusive or**. A disjunction is true when at least one of the two propositions is true. For instance, the inclusive or is being used in the statement

“Students who have taken calculus or computer science can take this class.”

Here, we mean that students who have taken both calculus and computer science can take the class, as well as the students who have taken only one of the two subjects. On the other hand, we are using the **exclusive or** when we say

“Students who have taken calculus or computer science, but not both, can enroll in this class.”

Here, we mean that students who have taken both calculus and a computer science course cannot take the class. Only those who have taken exactly one of the two courses can take the class.

Similarly, when a menu at a restaurant states, “Soup or salad comes with an entrée,” the restaurant almost always means that customers can have either soup or salad, but not both. Hence, this is an exclusive, rather than an inclusive, or.

EXAMPLE 6 What is the disjunction of the propositions p and q where p and q are the same propositions as in Example 5?

Solution: The disjunction of p and q , $p \vee q$, is the proposition

“Rebecca’s PC has at least 16 GB free hard disk space, or the processor in Rebecca’s PC runs faster than 1 GHz.”

This proposition is true when Rebecca’s PC has at least 16 GB free hard disk space, when the PC’s processor runs faster than 1 GHz, and when both conditions are true. It is false when both of these conditions are false, that is, when Rebecca’s PC has less than 16 GB free hard disk space and the processor in her PC runs at 1 GHz or slower.

DEFINITION 4 Let p and q be propositions. The *exclusive or* of p and q , denoted by $p \oplus q$, is the proposition that is true when exactly one of p and q is true and is false otherwise.

The truth table for the exclusive or of two propositions is displayed in Table 4.

TABLE 4 The Truth Table for the Exclusive Or of Two Propositions.		
p	q	$p \oplus q$
T	T	F
T	F	T
F	T	T
F	F	F

Conditional Statements

We will discuss several other important ways in which propositions can be combined.

DEFINITION 5 Let p and q be propositions. The *conditional statement* $p \rightarrow q$ is the proposition “if p , then q .” The conditional statement $p \rightarrow q$ is false when p is true and q is false, and true otherwise. In the conditional statement $p \rightarrow q$, p is called the *hypothesis* (or *antecedent* or *premise*) and q is called the *conclusion* (or *consequence*).

The statement $p \rightarrow q$ is called a conditional statement because $p \rightarrow q$ asserts that q is true on the condition that p holds. A conditional statement is also called an **implication**.

The truth table for the conditional statement $p \rightarrow q$ is shown in Table 5. Note that the statement $p \rightarrow q$ is true when both p and q are true and when p is false (no matter what truth value q has).

TABLE 5 The Truth Table for the Conditional Statement $p \rightarrow q$.		
p	q	$p \rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

Because conditional statements play such an essential role in mathematical reasoning, a variety of terminology is used to express $p \rightarrow q$. You will encounter most if not all of the following ways to express this conditional statement:

“if p , then q ”	“ p implies q ”
“if p , q ”	“ p only if q ”
“ p is sufficient for q ”	“a sufficient condition for q is p ”
“ q if p ”	“ q whenever p ”
“ q when p ”	“ q is necessary for p ”
“a necessary condition for p is q ”	“ q follows from p ”
“ q unless $\neg p$ ”	

A useful way to understand the truth value of a conditional statement is to think of an obligation or a contract. Consider a statement that a professor might make:

“If you get 100% on the final, then you will get an A.”

If you manage to get a 100% on the final, then you would expect to receive an A. If you do not get 100% you may or may not receive an A depending on other factors. However, if you do get 100%, but the professor does not give you an A, you will feel cheated.

We illustrate the translation between conditional statements and English statements in Example 7.

EXAMPLE 7 Let p be the statement “Maria learns discrete mathematics” and q the statement “Maria will find a good job.” Express the statement $p \rightarrow q$ as a statement in English.

Solution: From the definition of conditional statements, we see that when p is the statement “Maria learns discrete mathematics” and q is the statement “Maria will find a good job,” $p \rightarrow q$ represents the statement

“If Maria learns discrete mathematics, then she will find a good job.”

There are many other ways to express this conditional statement in English. Among the most natural of these are:

“Maria will find a good job when she learns discrete mathematics.”

“For Maria to get a good job, it is sufficient for her to learn discrete mathematics.”

and

“Maria will find a good job unless she does not learn discrete mathematics.”

Note that the way we have defined conditional statements is more general than the meaning attached to such statements in the English language. For instance, the conditional statement in Example 7 and the statement

“If it is sunny, then we will go to the beach.”

are statements used in normal language where there is a relationship between the hypothesis and the conclusion. Further, the first of these statements is true unless Maria learns discrete mathematics, but she does not get a good job, and the second is true unless it is indeed sunny, but we do not go to the beach. On the other hand, the statement

“If Juan has a smartphone, then $2 + 3 = 5$ ”

is true from the definition of a conditional statement, because its conclusion is true. (The truth value of the hypothesis does not matter then.) The conditional statement

“If Juan has a smartphone, then $2 + 3 = 6$ ”

is true if Juan does not have a smartphone, even though $2 + 3 = 6$ is false. We would not use these last two conditional statements in natural language (except perhaps in sarcasm), because there is no relationship between the hypothesis and the conclusion in either statement. The if-then construction used in many programming languages is different from that used in logic. Most programming languages contain statements such as **if p then S** , where p is a proposition and S is a program segment (one or more statements to be executed). When execution of a program encounters such a statement, S is executed if p is true, but S is not executed if p is false, as illustrated in Example 8.

BICONDITIONALS We now introduce another way to combine propositions that expresses that two propositions have the same truth value.

DEFINITION 6 Let p and q be propositions. The *bi-conditional statement* $p \leftrightarrow q$ is the proposition “ p if and only if q .” The bi-conditional statement $p \leftrightarrow q$ is true when p and q have the same truth values, and is false otherwise. Biconditional statements are also called *bi-implications*.

The truth table for $p \leftrightarrow q$ is shown in Table 6. Note that the statement $p \leftrightarrow q$ is true when both the conditional statements $p \rightarrow q$ and $q \rightarrow p$ are true and is false otherwise. That is why we use the words “if and only if” to express this logical connective and why it is symbolically written by combining the symbols \rightarrow and \leftarrow . There are some other common ways to express $p \leftrightarrow q$:

- “ p is necessary and sufficient for q ”
- “if p then q , and conversely”
- “ p iff q .”

The last way of expressing the bi-conditional statement $p \leftrightarrow q$ uses the abbreviation “iff” for “if and only if.” Note that $p \leftrightarrow q$ has exactly the same truth value as $(p \rightarrow q) \wedge (q \rightarrow p)$.

TABLE 6 The Truth Table for the Biconditional $p \leftrightarrow q$.		
p	q	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

EXAMPLE 10 Let p be the statement “You can take the flight,” and let q be the statement “You buy a ticket.” Then $p \leftrightarrow q$ is the statement

“You can take the flight if and only if you buy a ticket.”

This statement is true if p and q are either both true or both false, that is, if you buy a ticket and can take the flight or if you do not buy a ticket and you cannot take the flight. It is false when p and q have opposite truth values, that is, when you do not buy a ticket, but you can take the flight (such as when you get a free trip) and when you buy a ticket but you cannot take the flight (such as when the airline bumps you).

Truth Tables of Compound Propositions

We have now introduced four important logical connectives—conjunctions, disjunctions, conditional statements, and biconditional statements—as well as negations. We can use these connectives to build up complicated compound propositions involving any number of

propositional variables. We can use truth tables to determine the truth values of these compound propositions, as Example 11 illustrates. We use a separate column to find the truth value of each compound expression that occurs in the compound proposition as it is built up. The truth values of the compound proposition for each combination of truth values of the propositional variables in it is found in the final column of the table.

EXAMPLE 11 Construct the truth table of the compound proposition

$$(p \vee \neg q) \rightarrow (p \wedge q).$$

Solution: Because this truth table involves two propositional variables p and q , there are four rows in this truth table, one for each of the pairs of truth values TT, TF, FT, and FF. The first two columns are used for the truth values of p and q , respectively. In the third column we find the truth value of $\neg q$, needed to find the truth value of $p \vee \neg q$, found in the fourth column. The fifth column gives the truth value of $p \wedge q$. Finally, the truth value of $(p \vee \neg q) \rightarrow (p \wedge q)$ is found in the last column. The resulting truth table is shown in Table 7.

TABLE 7 The Truth Table of $(p \vee \neg q) \rightarrow (p \wedge q)$.					
p	q	$\neg q$	$p \vee \neg q$	$p \wedge q$	$(p \vee \neg q) \rightarrow (p \wedge q)$
T	T	F	T	T	T
T	F	T	T	F	F
F	T	F	F	F	T
F	F	T	T	F	F

Precedence of Logical Operators

We can construct compound propositions using the negation operator and the logical operators defined so far. We will generally use parentheses to specify the order in which logical operators in a compound proposition are to be applied. For instance, $(p \vee q) \wedge (\neg r)$ is the conjunction of $p \vee q$ and $\neg r$. However, to reduce the number of parentheses, we specify that the negation operator is applied before all other logical operators. This means that $\neg p \wedge q$ is the conjunction of $\neg p$ and q , namely, $(\neg p) \wedge q$, not the negation of the conjunction of p and q , namely $\neg(p \wedge q)$.

Another general rule of precedence is that the conjunction operator takes precedence over the disjunction operator, so that $p \wedge q \vee r$ means $(p \wedge q) \vee r$ rather than $p \wedge (q \vee r)$. Because this rule may be difficult to remember, we will continue to use parentheses so that the order of the disjunction and conjunction operators is clear.

Finally, it is an accepted rule that the conditional and bi-conditional operators \rightarrow and \leftrightarrow have lower precedence than the conjunction and disjunction operators, \wedge and \vee . Consequently, $p \vee q \rightarrow r$ is the same as $(p \vee q) \rightarrow r$. We will use parentheses when the order of the conditional operator and bi-conditional operator is at issue, although the conditional operator has

precedence over the bi-conditional operator. Table 8 displays the precedence levels of the logical operators, \neg , \wedge , \vee , \rightarrow , and \leftrightarrow .

TABLE 8 Precedence of Logical Operators.	
<i>Operator</i>	<i>Precedence</i>
\neg	1
\wedge	2
\vee	3
\rightarrow	4
\leftrightarrow	5

Logic and Bit Operations

Computers represent information using bits. A **bit** is a symbol with two possible values, namely, 0 (zero) and 1 (one). This meaning of the word bit comes from *binary digit*, because zeros and ones are the digits used in binary representations of numbers. The well-known statistician John Tukey introduced this terminology in 1946. A bit can be used to represent a truth value, because there are two truth values, namely, *true* and *false*. As is customarily done, we will use a 1 bit to represent true and a 0 bit to represent false. That is, 1 represents T (true), 0 represents F (false). A variable is called a **Boolean variable** if its value is either true or false. Consequently, a Boolean variable can be represented using a bit.

Computer **bit operations** correspond to the logical connectives. By replacing true by a one and false by a zero in the truth tables for the operators \wedge , \vee , and \oplus , the tables shown in Table 9 for the corresponding bit operations are obtained. We will also use the notation *OR*, *AND*, and *XOR* for the operators \vee , \wedge , and \oplus , as is done in various programming languages.

<i>Truth Value</i>	<i>Bit</i>
T	1
F	0

TABLE 9 Table for the Bit Operators <i>OR</i> , <i>AND</i> , and <i>XOR</i> .				
<i>x</i>	<i>y</i>	$x \vee y$	$x \wedge y$	$x \oplus y$
0	0	0	0	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	0

Information is often represented using bit strings, which are lists of zeros and ones. When this is done, operations on the bit strings can be used to manipulate this information.

DEFINITION 7 A *bit string* is a sequence of zero or more bits. The *length* of this string is the number of bits in the string.

EXAMPLE 12 101010011 is a bit string of length nine.

We can extend bit operations to bit strings. We define the **bitwise OR**, **bitwise AND**, and **bitwise XOR** of two strings of the same length to be the strings that have as their bits the *OR*, *AND*, and *XOR* of the corresponding bits in the two strings, respectively. We use the symbols \vee , \wedge , and \oplus to represent the bitwise *OR*, bitwise *AND*, and bitwise *XOR* operations, respectively.

We illustrate bitwise operations on bit strings with Example 13.

EXAMPLE 13 Find the bitwise *OR*, bitwise *AND*, and bitwise *XOR* of the bit strings 01 1011 0110 and 11 0001 1101. (Here, and throughout this book, bit strings will be split into blocks of four bits to make them easier to read.)

Solution: The bitwise *OR*, bitwise *AND*, and bitwise *XOR* of these strings are obtained by taking the *OR*, *AND*, and *XOR* of the corresponding bits, respectively. This gives us

```
01 1011 0110
11 0001 1101
11 1011 1111 bitwise OR
01 0001 0100 bitwise AND
10 1010 1011 bitwise XOR
```

Exercise 1.1

1. Which of these sentences are propositions? What are the truth values of those that are propositions?
 - a) Boston is the capital of Massachusetts.
 - b) Miami is the capital of Florida.
 - c) $2 + 3 = 5$.
 - d) $5 + 7 = 10$.
 - e) $x + 2 = 11$.
 - f) Answer this question.

2. Which of these are propositions? What are the truth values of those that are propositions?
 - a) Don't go.
 - b) What time is it?
 - c) There are no black flies in Maine.
 - d) $4 + x = 5$.
 - e) The moon is made of green cheese.
 - f) $2n \geq 100$.

3. What is the negation of each of these propositions?
 - a) Mei has an MP3 player.
 - b) There is no pollution in New Jersey.
 - c) $2 + 1 = 3$.
 - d) The summer in Maine is hot and sunny.

4. Suppose that Smartphone A has 256 MB RAM and 32GB ROM, and the resolution of its camera is 8 MP; Smartphone B has 288 MB RAM and 64 GB ROM, and the resolution of its camera is 4 MP; and Smartphone C has 128 MB RAM and 32 GB ROM, and the resolution of its camera is 5 MP. Determine the truth value of each of these propositions.
 - a) Smartphone B has the most RAM of these three smartphones.
 - b) Smartphone C has more ROM or a higher resolution camera than Smartphone B.
 - c) Smartphone B has more RAM, more ROM, and a higher resolution camera than Smartphone A.

- d) If Smartphone B has more RAM and more ROM than Smartphone C, then it also has a higher resolution camera.
- e) Smartphone A has more RAM than Smartphone B if and only if Smartphone B has more RAM than Smartphone A.

5. Let p and q be the propositions

p : I bought a lottery ticket this week.

q : I won the million dollar jackpot.

Express each of these propositions as an English sentence.

a) $\neg p$

b) $p \vee q$

c) $p \rightarrow q$

d) $p \wedge q$

e) $p \leftrightarrow q$

6. Let p and q be the propositions "Swimming at the New Jersey shore is allowed" and "Sharks have been spotted near the shore," respectively. Express each of these compound propositions as an English sentence.

a) $\neg q$

b) $p \wedge q$

c) $\neg p \vee q$

d) $p \rightarrow \neg q$

7. Let p and q be the propositions

p : It is below freezing.

q : It is snowing.

Write these propositions using p and q and logical connectives (including negations).

a) It is below freezing and snowing.

b) It is below freezing but not snowing.

c) It is not below freezing and it is not snowing.

d) It is either snowing or below freezing (or both).

e) If it is below freezing, it is also snowing.

8. Determine whether these biconditionals are true or false.
- $2 + 2 = 4$ if and only if $1 + 1 = 2$.
 - $1 + 1 = 2$ if and only if $2 + 3 = 4$.
 - $1 + 1 = 3$ if and only if monkeys can fly.
 - $0 > 1$ if and only if $2 > 1$.
9. Determine whether each of these conditional statements is true or false.
- If $1 + 1 = 2$, then $2 + 2 = 5$.
 - If $1 + 1 = 3$, then $2 + 2 = 4$.
 - If $1 + 1 = 3$, then $2 + 2 = 5$.
 - If monkeys can fly, then $1 + 1 = 3$.
 - If $1 + 1 = 3$, then dogs can fly.
 - If $1 + 1 = 2$, then dogs can fly.
10. How many rows appear in a truth table for each of these compound propositions?
- $p \rightarrow \neg p$
 - $(p \vee \neg r) \wedge (q \vee \neg s)$
 - $(q \rightarrow \neg p) \vee (\neg p \rightarrow \neg q)$
 - $(p \vee \neg t) \wedge (p \vee \neg s)$
11. Construct a truth table for each of these compound propositions.
- $p \wedge \neg p$
 - $p \vee \neg p$
 - $(p \vee \neg q) \rightarrow q$
 - $(p \vee q) \rightarrow (p \wedge q)$
 - $(p \rightarrow q) \rightarrow (q \rightarrow p)$

12. Find the bitwise OR, bitwise AND, and bitwise XOR of each of these pairs of bit strings.

a) 101 1110, 010 0001

b) 1111 0000, 1010 1010

c) 00 0111 0001, 10 0100 1000

d) 11 1111 1111, 00 0000 0000

Applications of Propositional Logic (Self–Reading)

Introduction

Logic has many important applications to mathematics, computer science, and numerous other disciplines. Statements in mathematics and the sciences and in natural language often are imprecise or ambiguous. To make such statements precise, they can be translated into the language of logic. For example, logic is used in the specification of software and hardware. Furthermore, propositional logic and its rules can be used to design computer circuits, to construct computer programs, to verify the correctness of programs, and to build expert systems. Logic can be used to analyze and solve many familiar puzzles.

Translating English Sentences

There are many reasons to translate English sentences into expressions involving propositional variables and logical connectives. In particular, English (and every other human language) is often ambiguous. Translating sentences into compound statements removes the ambiguity. Once we have translated sentences from English into logical expressions we can analyze these logical expressions to determine their truth values.

To illustrate the process of translating an English sentence into a logical expression, consider Examples 1.

EXAMPLE 1 How can this English sentence be translated into a logical expression?

“You can access the Internet from campus only if you are a computer science major or you are not a freshman.”

Solution: Let a , c , and f represent “You can access the Internet from campus,” “You are a computer science major,” and “You are a freshman,” respectively. Noting that “only if” is one way a conditional statement can be expressed, this sentence can be represented as

$$a \rightarrow (c \vee \neg f).$$

Boolean Searches

Logical connectives are used extensively in searches of large collections of information, such as indexes of Web pages. Because these searches employ techniques from propositional logic, they are called **Boolean searches**.

In Boolean searches, the connective *AND* is used to match records that contain both of two search terms, the connective *OR* is used to match one or both of two search terms, and the connective *NOT* (sometimes written as *AND NOT*) is used to exclude a particular search term.

EXAMPLE Web Page Searching

For instance, using Boolean searching to find Web pages about universities in New Mexico, we can look for pages matching *NEW AND MEXICO AND UNIVERSITIES*. The results of this search will include those pages that contain the three words *NEW*, *MEXICO*, and *UNIVERSITIES*. This will include all of the pages of interest, together with others such as a page about new universities in Mexico. (Note that in Google, and many other search engines, the word “AND” is not needed, although it is understood, because all search terms are included by default.