



# Chapter 1 : The Foundation :

## Logic and proofs

### Lecture 1:

1. Propositional Logic
2. Applications of Propositional Logic

### Prepared by:

- Dr. Abbas Rammal
- Dr. Rabih Assaf

# 1.1

## Propositional Logic

### 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.

We use letters to denote **propositional variables** (or **statement variables**), that is, variables that represent propositions, just as letters are used to denote numerical variables.

**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.”

Table 1 displays the **truth table** for the negation of a proposition  $p$ .

<b>TABLE 1 The Truth Table for the Negation of a Proposition.</b>	
$p$	$\neg p$
T	F
F	T

## 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  $p \wedge q$ .

<b>TABLE 2</b> 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

## 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  $p \vee q$ .

<b>TABLE 3</b> 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

#### 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.

<b>TABLE 4</b> 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

## 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.

**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

The following ways to express this conditional statement:

“if  $p$ , then  $q$ ”

“if  $p$ ,  $q$ ”

“ $p$  is sufficient for  $q$ ”

“ $q$  if  $p$ ”

“ $q$  when  $p$ ”

“a necessary condition for  $p$  is  $q$ ”

“ $q$  unless  $\neg p$ ”

“ $p$  implies  $q$ ”

“ $p$  only if  $q$ ”

“a sufficient condition for  $q$  is  $p$ ”

“ $q$  whenever  $p$ ”

“ $q$  is necessary for  $p$ ”

“ $q$  follows from  $p$ ”

**CONVERSE, CONTRAPOSITIVE, AND INVERSE** We can form some new conditional statements starting with a conditional statement  $p \rightarrow q$ . In particular, there are three related conditional statements that occur so often that they have special names. The proposition  $q \rightarrow p$  is called the **converse** of  $p \rightarrow q$ . The **contrapositive** of  $p \rightarrow q$  is the proposition  $\neg q \rightarrow \neg p$ . The proposition  $\neg p \rightarrow \neg q$  is called the **inverse** of  $p \rightarrow q$ . We will see that of these three conditional statements formed from  $p \rightarrow q$ , only the contrapositive always has the same truth value as  $p \rightarrow q$ .

## DEFINITION 6

Let  $p$  and  $q$  be propositions. The *biconditional statement*  $p \leftrightarrow q$  is the proposition “ $p$  if and only if  $q$ .” The biconditional 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*.

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  $q$ .”

**EXAMPLE**

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. 

<b>TABLE 7</b> 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

# 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.

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.

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

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.

**TABLE 9** Table for the Bit Operators *OR*, *AND*, and *XOR*.

$x$	$y$	$x \vee y$	$x \wedge y$	$x \oplus y$
0	0	0	0	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	0

## 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

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.