Session 2: The Foundations – Logic and Proofs

Objectives

By the end this lesson, the learner should be able to:

- Explain Proposition Logic and its applications
- Draw Truth Tables
- Explain the nature mathematical Proofs
- Identify various types of proof

2.1 Introduction to Logic and Proofs

The rules of logic specify the meaning of mathematical statements.
For instance, these rules help us understand and reason with statements such as:

- "There exists an integer that is not the sum of two squares" and
- "For every positive integer n, the sum of the positive integers not exceeding n is $n(n + 1)/2$."

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, and
- to other areas of computer science, as well as to many other fields of study.

To understand mathematics, we must understand what makes up a correct mathematical argument, that is, *a **proof.*** Once we prove a mathematical statement is true, we call it a ***theorem.***

A collection of theorems are organized in a topic. To learn a mathematical topic, a person needs to actively construct mathematical arguments on a topic, and not just read exposition.

Moreover, knowing the proof of a theorem often makes it possible to modify the result to fit new situations.

Everyone knows that proofs are important throughout mathematics, but many people find it surprising how important proofs are in computer science.

In fact, proofs are used:

- to verify that computer programs produce the correct output for all possible input values,
- to show that algorithms always produce the correct result, to establish the security of a system, and
- to create artificial intelligence. Furthermore, automated reasoning systems have been created to allow computers to construct their own proofs.

In this Topic, we will explain what makes up a correct *mathematical argument* and introduce tools to construct these arguments.
We will develop an arsenal of different proof methods that will enable us to prove many different types of results.
After introducing many different methods of proof, we will introduce several strategies for constructing proofs.

2.2 **Propositional Logic**

The rules of logic give precise meaning to mathematical statements. These rules are used to distinguish between valid and invalid mathematical arguments.

Because a major goal of this Course Unit is to teach the student how to understand and construct correct mathematical arguments, we begin our study of discrete mathematics with an introduction to logic.

Logic has numerous applications to computer science. The rules of logic are used in the design of computer circuits, the construction of computer programs, the verification of the correctness of programs, and in many other ways.

Definition: Proposition

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. Nairobi is the capital city of Kenya.
2. Pigeon is a is mammal.
3. 1 + 1 = 2.
4. 2 + 2 = 3.

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

EXAMPLE 2

Some sentences that are not propositions. 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.

2.3 Propositional variables

We use letters to denote **propositional variables** (or **sentential variables**), that is, variables that represent propositions, just as letters are used to denote numerical variables. The conventional letters used for propositional variables are $p, q, r, s, \dots$.

2.4 Truth value

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*. Propositions that cannot be expressed in terms of simpler propositions are called **atomic propositions**.

Note:

New propositions, called **compound propositions**, are formed from existing propositions using logical operators.

Definition: Negation of p

Let p be a proposition. The *negation of p*, denoted by ¬p (also denoted by p), is the statement "It is not the case that p."

The proposition ¬p is read "not p."

Note:

The *truth value* of the negation of p, ¬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 "Kariuki's smartphone has at least 32 GB of memory" and express this in simple English.

Solution:

The negation is "It is not the case that Kariuki's smartphone has at least 32 GB of memory."

This negation can also be expressed as "Kariuki's smartphone does not have at least 32 GB of memory"

or even more simply as "Kariuki's smartphone has less than 32 GB of memory."

Truth Table for the Negation of a Proposition

The table has a row for each of the two possible truth values of p. Each row shows the truth value of $\neg p$ corresponding to the truth value of p for this row.

Negation of a Proposition
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.

P	**¬p**
T	F
F	T

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

2.5 Definition: Connectives – Conjunction and Disjunction
Are logical operators that are used to form new propositions from two or more existing propositions. The connectives are **Conjunction and Disjunction**

Conjunction

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.

Truth Table of $p \wedge q$

P	q	p∧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.

Note
The word "but" sometimes is used instead of "and" in a conjunction. EG. "The sun is shining, but it is raining" is another way of saying "The sun is shining and it is raining."

Disjunction
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**.

The Truth Table Disjunction

P	q	p ∨ 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. That is, $p \vee q$ is true when both p and q are **true** or when exactly one of p and q is **true.**

EXAMPLE 6 Translate the statement "Students who have taken calculus or introductory computer science can take this class" in a statement in propositional logic using the propositions p: "A student who has taken calculus can take this class" and q: "A student who has taken introductory computer science can take this class."

Solution: We assume that this statement means that students who have taken both calculus and introductory computer science can take the class, as well as the students who have taken only one of the two subjects. Hence, this statement can be expressed as $p \vee q$, the inclusive or, or disjunction, of p and q.

Exclusive OR

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

P	q	P ⊕ q
T	T	F
T	F	T
F	T	T
F	F	F

EXAMPLE 8

Let p and q be the propositions that state "A student can have a salad with dinner" and "A student can have soup with dinner," respectively. What is $p \oplus q$, the exclusive or of p and q?

Solution: The exclusive or of p and q is the statement that is true when exactly one of p and q is true. That is, $p \oplus q$ is the statement "A student can have soup or salad, but not both, with dinner." Note that this is often stated as "A student can have soup or a salad with dinner," without explicitly stating that taking both is not permitted. ◂

EXAMPLE 9

Express the statement "I will use all my savings to travel to Europe or to buy an electric car" in propositional logic using the statement p: "I will use all my savings to

travel to Europe" and the statement q: "I will use all my savings to buy an electric car."
Solution:

To translate this statement, we first note that the or in this statement must be an exclusive or because this student can either use all his or her savings to travel to Europe or use all these savings to buy an electric car, but cannot both go to Europe and buy an electric car. (This is clear because either option requires all his savings.) Hence, this statement can be expressed as $p \oplus q$.

2.6 Definition: Conditional Statements

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

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).

Truth Table for the Conditional Statement $p \rightarrow q$.

P	q	P → 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 $p \to q$. 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 .p" "q provided that p"

A useful way to understand the truth value of a conditional statement is to think of an obligation
or a contract.
For example,
a) The pledge many politicians make when running for office is
"If I am elected, then I will lower taxes."

b) A professor might make
"If you get 100% on the final, then you will get an A."

EXAMPLE 10
Let p be the statement "Patricia learns discrete mathematics" and q the statement "Patricia will find a good job." Express the statement $p \to q$ as a statement in English. Hence draw a Truth Table

Solution:
From the definition of conditional statements, we see that when p is the statement

"Patricia learns discrete mathematics" and q is the statement "Patricia will find a good job," $p \to q$ represents the statement
"If Patricia 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:
"Patricia will find a good job when she learns discrete mathematics."
"For Patricia to get a good job, it is sufficient for her to learn discrete mathematics."
and
"Patricia will find a good job unless she does not learn discrete mathematics."

Further examples:

Express the statement $p \to q$ as a statement in English:
"If it is sunny, then we will go to the beach"
"If Juan has a smartphone, then $2 + 3 = 5$"

EXAMPLE 11 What is the value of the variable x after the statement

if $2 + 2 = 4$ **then** $x := x + 1$

if $x = 0$ before this statement is encountered? (The symbol := stands for assignment. The statement $x := x + 1$ means the assignment of the value of $x + 1$ to x.)

Solution: Because $2 + 2 = 4$ is true, the assignment statement $x := x + 1$ is executed. Hence, x has the value $0 + 1 = 1$ after this statement is encountered.

2.7 CONVERSE, CONTRAPOSITIVE, AND INVERSE

We can form some new conditional statements starting with a conditional statement $p \to q$. In particular, there are three related conditional statements that occur so often that they have special names.

Converse
The proposition $q \to p$ is called the **converse** of $p \to q$.

Contrapositive
The **contrapositive** of $p \to q$ is the proposition $\neg q \to \neg p$. The proposition $\neg p \to \neg q$ is called the **inverse** of $p \to q$.

These three conditional statements formed from $p \to q$,

Exercise:
Show that the contrapositive always has the same truth value as $p \to q$.

Example:
Find the contrapositive, the converse, and the inverse of the conditional statement "The home team wins whenever it is raining."

Solution:
Because "q whenever p" is one of the ways to express the conditional statement $p \to q$, the original statement can be rewritten as "If it is raining, then the home team wins."

The **contrapositive** of this conditional statement is "If the home team does not win, then it is not raining."

The **converse** is "If the home team wins, then it is raining."

The **inverse** is "If it is not raining, then the home team does not win."

Only the contrapositive is equivalent to the original statement.

Truth Tables of Compound Propositions

We have now introduced five important logical connectives—conjunction, disjunction, exclusive or, and implication.

EXAMPLE

Construct the truth table of the compound proposition $(p \vee \neg q) \to (p \wedge q)$.

Solution: Since 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 $\vee q$, needed to find the truth value of $p \vee .q$, found in the fourth column. The fifth column gives the truth value of $p \wedge q$. Finally, the truth value of $(p \vee \neg q) \to (p \wedge q)$ is found in the last column. The resulting truth table is shown in Table below.

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

p	q	$\neg q$	$p \vee \neg q$	$p \wedge q$	$(p \vee \neg q) \to (p \wedge q)$
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 $\neg(p \wedge q)$. (It is generally the case that unary operators that involve only one object precede binary operators.)

Another general rule of precedence is that the conjunction operator takes precedence over the disjunction operator, so that $p \vee q \wedge r$ means $p \vee (q \wedge r)$ rather than $(p \vee q) \wedge r$ and $p \wedge q \vee r$ 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 operators, $\rightarrow$,have lower precedence than the conjunction and disjunction operators, $\wedge$ and $\vee$. Consequently, $p \rightarrow q \vee r$ means $p \rightarrow (q \vee r)$ rather than $(p \rightarrow q) \vee r$ and $p \vee q \rightarrow r$ means $(p \vee q) \rightarrow r$ rather than $p \vee (q \rightarrow r)$.

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 biconditional operator.

Table below displays the precedence levels of the logical operators, $\neg$, $\wedge$, $\vee$, and $\rightarrow$.

TABLE Precedence of Logical Operators.

Operat	*Precedence*
$\neg$	1
$\wedge$	2
$\vee$	3
$\rightarrow$	4

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 **b**inary dig***it,*** 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*.

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).

Truth Value Bit

T	**1**
F	**0**

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 ∧, ∨, and ⊕, the columns in Table for ***Truth Value Bit,*** the corresponding bit operations are obtained. We will also use the notation *OR*, *AND*, and *XOR* for the operators ∨, ∧, and ⊕, as is done in various programming languages.

TABLE 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

Bit Strings

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.

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

EXample101010011 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 ∨, ∧, and ⊕ to represent the bitwise *OR*, bitwise *AND*, and bitwise *XOR* operations, respectively.

Example:
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*

Exercises

10. 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) .*p* **b)** *p* ∨ *q* **c)** *p* → *q*

d) *p* ∧ *q* **e)** *p* ↔ *q* **f)** .*p* → .*q*

g) .*p* ∧ .*q* **h)** .*p* ∨ (*p* ∧ *q*)

11. 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) .*q* **b)** *p* ∧ *q* **c)** .*p* ∨ *q*

d) *p* → .*q* **e)** .*q* → *p* **f)** .*p* → .*q*

g) *p* ↔ .*q* **h)** .*p* ∧ (*p*∨ .*q*)

12. Let *p* and *q* be the propositions "The election

47. 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

48. Evaluate each of these expressions.

a) $1\ 1000 \wedge (0\ 1011 \vee 1\ 1011)$
b) $(0\ 1111 \wedge 1\ 0101) \vee 0\ 1000$
c) $(0\ 1010 \oplus 1\ 1011) \oplus 0\ 1000$
d) $(1\ 1011 \vee 0\ 1010) \wedge (1\ 0001 \vee 1\ 1011)$