Unit 1: 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”.

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. 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.

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. The logic, on the basis of its representation capability, is of two types: Propositional Logic and Predicate Logic.

Propositions and Truth Functions:

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

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.

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 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, …. 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.

Atomic Proposition and Compound Proposition:

Propositions that cannot be expressed in terms of simpler propositions are called atomic propositions. Its truth value does not depend on another proposition. These are also known as primitive propositions and building blocks of propositional logic.

e.g., Kathmandu is the capital city of Nepal.
C is a structural programming language.

A compound proposition is a proposition that can be divided into simpler, atomic propositions. Compound propositions are formed by composing existing atomic propositions using logical operators.

e.g., Computers are diligent and versatile.
If it rains, then I will go to sleep.

Truth Function and Truth Table:

In logic, a truth function is a function that accepts truth value as input and produces a truth value as output. Truth functions are sometimes called Boolean Functions. Truth table is a table which consists of all possible input and output truth value of any proposition. Truth tables are especially valuable in the determination of the truth values of propositions, constructed from simple propositions.

Propositional Logic:

The area of logic that deals with propositions is called propositional logic. Propositional logic, also known as sentential logic, is the branch of logic that studies way of joining or modifying propositions to form more complicated propositions as well as logical relationships. These rules are used to distinguish between valid and invalid mathematical arguments. Besides the importance of logic in understanding mathematical reasoning, logic has numerous applications to computer science.

Applications of Propositional Logic:

The applications of propositional logic are as follows:

1. Inference and Reasoning: A new proposition can be derived from the set of interrelated propositions.
2. Translation of English Sentences: English Sentences can be ambiguous and this ambiguity can lead to faulty decision making. To remove such ambiguity, we can translate English sentences into logical expression with the help of propositional logic.
3. System Specifications: Translating sentences in natural language (such as English) into logical expressions is an essential part of specifying both hardware and software systems. System and software engineers take requirements in natural language and produce precise and unambiguous specifications that can be used as the basis for system development.
4. 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, Links 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 is used to exclude a particular search term.
5. Logic Puzzles: Puzzles that can be solved using logical reasoning are known as logic puzzles. Solving logic puzzles is an excellent way to practice working with the rules of logic. Also, computer programs designed to carry out logical reasoning often use well-known logic puzzles to illustrate their capabilities.

6. Logic Circuits: Propositional logic can be applied to the design of computer hardware. A logic circuit (or digital circuit) receives input signals $p_1, p_2, \dots, p_n$, each a bit [either 0 (off) or 1 (on)], and produces output signals $s_1, s_2, \dots, s_n$, each a bit.
7. Artificial Intelligence: Artificial intelligence makes use of fuzzy logic (computing based on degrees of truth rather than Boolean true or false), and propositional logic can be used to present the fuzzy statements.

Logical Operators or Connectives:

Many mathematical statements are constructed by combining one or more propositions. New propositions, called compound propositions, are formed from existing propositions using logical operators. There are multiple logical operations such as NOT, AND, OR, XOR etc.

Negation (NOT):

Let p be a proposition. The negation of p, denoted by ¬ p, is the statement

"It is not the case that p."

The proposition ¬ p is read "not p." The truth value of the negation of p, ¬ p, is the opposite of the truth value of p. Other notations are: ~p, −p, p′, Np, and !p.

e.g.,

Negation of the proposition "I love dogs" is "I do not love dogs". If the sentence "I love dogs" is denoted by p, then its negation "I do not love dogs" is denoted by ¬ p.

The negation of the proposition "Michael's PC runs Linux" is "It is not the case that Michael's PC runs Linux." or "Michael's PC does not run Linux."

Truth Table:

p	**¬ p**
T	F
F	T

Conjunction (AND):

Let p and q be propositions. The conjunction of p and q, denoted by p ∧ q, is the proposition "p and q." The conjunction p ∧ q is true when both p and q are true, and is false otherwise.

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

e.g.,

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." is "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."

Disjunction (OR):

Let p and q be propositions. The disjunction of p and q, denoted by p ∨ q, is the proposition "p or q." The disjunction p ∨ q is false when both p and q are false and is true otherwise.

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

e.g.,

The disjunction 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." is "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.

Exclusive OR (XOR):

Let p and q be propositions. The exclusive or of p and q, denoted by p ⊕ 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	$p \oplus q$
T	T	F
T	F	T
F	T	T
F	F	F

e.g.,

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 ⊕ q, the exclusive or of p and q?

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 ⊕ 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.

Conditional Statement/Implication (→):

Let p and q be propositions. The conditional statement p → q is the proposition "if p, then q." The conditional statement p → q is false when p is true and q is false, and true otherwise. In the conditional statement p → q, p is called the hypothesis (or antecedent or premise) and q is called the conclusion (or consequence).

The statement p → q is called a conditional statement because p → 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 → q is shown as:

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 $p \rightarrow q$.

We 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”

For example, the pledge many politicians make when running for office is

“If I am elected, then I will lower taxes.”

If the politician is elected, voters would expect this politician to lower taxes. Furthermore, if the politician is not elected, then voters will not have any expectation that this person will lower taxes, although the person may have sufficient influence to cause those in power to lower taxes. It is only when the politician is elected but does not lower taxes that voters can say that the politician has broken the campaign pledge. This last scenario corresponds to the case when p is true but q is false in $p \rightarrow q$.

Similarly, 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 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.

Converse, Contrapositive and Inverse:

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

- The proposition q → p is called the converse of p → q.
- The contrapositive of p → q is the proposition ¬q → ¬p.
- The proposition ¬p → ¬q is called the inverse of p → q.

Inverse of Implication:

When we add 'not' to the hypothesis and conclusion of implication p → q, then it becomes ¬p → ¬q, which is known as inverse of p → q.

e.g.,

If it is raining, then the road is muddy.

Inverse: If it is not raining, then the road is not muddy.

Converse of Implication:

When we interchange/flip the hypothesis and conclusion of implication p → q, then the result becomes q → p, which is known as the converse of given implication.

e.g.,

If it is raining, then the road is muddy.

Converse: If the road is muddy, then it is raining.

Contrapositive of Implication:

When we interchange/flip the hypothesis and conclusion of inverse statement of implication p → q, then the resulting statement ¬q → ¬p is known as the contrapositive.

e.g.,

If it is raining, then the road is muddy.

Inverse: If it is not raining, then the road is not muddy.

Contrapositive: If the road is not muddy, then it is not raining.

Examples:

Find the inverse, converse and contrapositive of the following statements.

1. If two angles are congruent, then they have same measures.
    a. Statement (p → q): If two angles are congruent, then they have same measures.
    b. Inverse (¬p → ¬q): If two angles are not congruent, then they do not have same measures.
    c. Converse (q → p): If two angles have the same measures, then they are congruent.
    d. Contrapositive (¬q → ¬p): If two angles do not have same measures, then they are not congruent.
2. The home team wins whenever it is raining.
    a. Statement (p → q): If it is raining, then the home team wins.
    b. Inverse (¬p → ¬q): If it is not raining, then the home team does not win.
    c. Converse (q → p): If the home team wins, then it is raining.
    d. Contrapositive (¬q → ¬p): If the home team does not win, then it is not raining.
3. If quadrilateral is rectangle, then each of its angles are 90 degrees.
    a. Statement (p → q): If quadrilateral is rectangle, then each of its angles are 90 degrees.
    b. Inverse (¬p → ¬q): If quadrilateral is not rectangle, then each of its angles are not 90 degrees.
    c. Converse (q → p): If quadrilateral has each of its angles 90 degrees, then it is a rectangle.
    d. Contrapositive (¬q → ¬p): If the quadrilateral does not have each of its angles 90 degrees, then it is not a rectangle.

Biconditional (↔):

Let p and q be propositions. The biconditional statement p ↔ q is the proposition "p if and only if q." The biconditional statement p ↔ q is true when p and q have the same truth values, and is false otherwise. Biconditional statements are also called bi-implications.

Note that the statement p ↔ q is true when both the conditional statements p → q and q → 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 → and ←. There are some other common ways to express p ↔ q:

- "p is necessary and sufficient for q"
- "if p then q, and conversely"
- "p iff q."
- "p exactly when q."

p	q	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

e.g.,

Let p be the statement "You can take the flight," and let q be the statement "You buy a ticket." Then p ↔ 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).

Representing English Sentences in Propositional Logic:

Translation of English sentence into propositional logic has much importance. In particular, English and every other human language is often ambiguous. Translating sentences into compound statements removes the ambiguity. Moreover, once we translate sentences from English into logical expressions, we can analyze these logical expressions to determine their truth values, we can manipulate them and we can use rules of inference to reason about them.

We can also use rule of inference on these logical expressions to infer or derive new expressions.

The process of translating English sentences into logical expression consists of following steps:

1. First, we break down the given complex English sentence into atomic sentences.
2. We represent each atomic sentence by propositional variables.

3. Each atomic sentence is connected with appropriate connectives.

e.g.,

If it is raining, the home team wins the game.

Let's break this sentence into two atomic sentences as:

- It is raining
- Home team wins the game

Let p=" It is raining" and q=" Home team wins the game". In this example, the sentences are connected with implication. So, the logical expression will be $p \rightarrow q$.

<u>Example 1:</u>

Translate the following declarative sentences:

p: It is raining.

q: Sita is sick.

r: Ram stayed up late last night.

t: Kathmandu is the capital of Nepal.

u: Ashok is a loud mouth.

a. Translating Negation:
   i. $\neg p$: It is not raining, where p: It is raining.
   ii. $\neg (\neg q)$: It is not the case that Sita isn't sick.

      Since q: Sita is sick, $\neg q$: Sita isn't sick, and $\neg (\neg q)$: It is not the case that Sita isn't sick.
b. Translating Conjunction:
   i. $p \land q$: It is raining and Sita is sick.
   ii. $\neg t \land \neg p$: Kathmandu isn't the capital of Nepal and it isn't raining.
   iii. $\neg p \land \neg q$: Its isn't raining and Sita isn't sick.

      Alternatively, it is not the case that both it is raining and Sita is sick.
   iv. $\neg p \land q$: It isn't raining and Sita is sick.
c. Translating Disjunction:
   i. $t \lor p$: Kathmandu is capital city of Nepal or it is raining.
   ii. $\neg t \lor \neg p$: Kathmandu isn't the capital of Nepal or it isn't raining.

iii. (t ∧ p) ∨ q: Kathmandu is the capital of Nepal and it is raining or Sita is sick.

iv. r ∨ ¬ r: Ram stayed up late last night or didn't stay up late last night.

v. r ∨ ¬ u: Ram stayed up late last night or Ashok isn't a loud mouth.

d. Translating Implication:

i. p → q: If it is raining, then Sita is sick.

ii. u → q: Sita is sick when Ashok is a loudmouth.

iii. (t ∧ p) → q: If Kathmandu is the capital of Nepal and it is raining, then Sita is sick. Alternatively, Kathmandu is the capital of Nepal and it is raining implies that Sita is sick.

e. Translating Biconditional:

i. p ↔ q: It is raining if and only if Sita is sick.

ii. t ↔ r: Kathmandu is the capital of Nepal is equivalent to Ram stayed up late last night.

<u>Example 2:</u>

Let p, q and r be the propositions defined as:

p: Ram gets A on final exam.

q: Ram does every exercise on the book.

r: Ram is the topper of this class.

Translate the following into logical expression:

a. Ram is the topper of this class but Ram does not do every exercise on the book.

Solution: r → ¬ q

b. Ram gets A on final exam; Ram does every exercise on the book and Ram is the topper of this class.

Solution: p ∧ q ∧ r

c. To be topper of this class, it is necessary for Ram to get A on final exam.

Solution: We can rephrase it as: If Ram gets A on final exam, then he will be topper of this class.

p → r

d. Getting A on final exam and doing every exercise on the book is sufficient for Ram to be topper of this class.

   Solution: $(p \land q) \rightarrow r$

<u>Example 3:</u>

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: There are many ways to translate this sentence into a logical expression. Although it is possible to represent the sentence by a single propositional variable, such as p, this would not be useful when analyzing its meaning or reasoning with it. Instead, we will use propositional variables to represent each sentence part and determine the appropriate logical connectives between them. In particular, we let p, q, and r 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 $q \lor \neg r \rightarrow p$.

<u>Example 4:</u>

Express the specification "The automated reply cannot be sent when the file system is full" using logical connectives.

Solution: One way to translate this is to let p denote "The automated reply can be sent" and q denote "The file system is full." Then $\neg p$ represents "It is not the case that the automated reply can be sent," which can also be expressed as "The automated reply cannot be sent." Consequently, our specification can be represented by the conditional statement $q \rightarrow \neg p$.

<u>Example 5:</u>

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

"You cannot ride the roller coaster if you are under 4 feet tall unless you are older than 16 years old."

Solution: It can be rephrased as: "Unless you are over 16 years old, you must be taller than 4 feet to ride the roller coaster." or "In order to ride the roller coaster, you must be both taller than 4 feet and older than 16 years old."

Let q, r, and s represent “You can ride the roller coaster,” “You are under 4 feet tall,” and “You are older than 16 years old,” respectively. Then the sentence can be translated to

$r \land \neg s \rightarrow \neg q$.

Example 6:

Translate the following paragraph into logical expression:

“Whenever the system software is being upgraded, users cannot access the file system. If users can access the file system, then they can save new files. If users cannot save new files, then system software is not being upgraded.”

Solution: Let p: The system software is being upgraded, q: Users can access the file system and r: Users can save new files.

Now,

$p \rightarrow \neg q$: Whenever the system software is being upgraded, users cannot access the file system.

$q \rightarrow r$: If users can access the file system, then they can save new files.

$\neg q \rightarrow \neg r$: If users cannot save new files, then system software is not being upgraded.

So, the whole paragraph can be written as:

$(p \rightarrow \neg q) \land (q \rightarrow r) \land (\neg q \rightarrow \neg r)$

Example 7:

Translate the given logical expressions into English sentence

a. $\neg p$ b. $r \land \neg p$ c. $\neg r \lor p \lor q$

when, p: It rained last night. q: The sprinkles came last night. r: The lawn was wet this morning.

Solution:

a. $\neg p$: It did not rain last night.
b. $r \land \neg p$: The lawn was wet this morning and it didn’t rain last night.
c. $\neg r \lor p \lor q$: Either the lawn was not wet this morning or it rained last night or the sprinkles came last night.

Note: To translate English sentences to the proposition symbolic form, follow these steps:

- Restate the given sentence into building block sentences.

- Give the symbol to each sentence, and
- Substitute the symbols using connectives.

Propositional Equivalences:

An important type of step used in a mathematical argument is the replacement of a statement with another statement with the same truth value. Because of this, methods that produce propositions with the same truth value as a given compound proposition are used extensively in the construction of mathematical arguments. Note that we will use the term "compound proposition" to refer to an expression formed from propositional variables using logical operators, such as $p \land q$.

Classification of compound propositions is done according to their possible truth values. A compound proposition that is always true, no matter what the truth values of the propositional variables that occur in it, is called a tautology. A compound proposition that is always false is called a contradiction. A compound proposition that is neither a tautology nor a contradiction is called a contingency.

Tautology, Contradiction and Contingency:

Tautology:

A compound proposition that is always true, no matter what the truth values of the propositional variables that occur in it, is called a tautology.

Example 1:

Show that $p \lor \neg p$ is tautology.

Solution:

We can verify this statement with help of truth table as follows:

Truth Table:

p	**¬p**	**p ∨ ¬p**
T	F	T
F	T	T

Here, the last column consists of all true truth values. So, p ∨ ¬p is always true and hence is a tautology.

Example 2:

Show that ¬ (p → q) → ¬q is tautology using truth table.

Solution:

We can verify this statement with help of truth table as follows:

Truth Table:

p	**q**	**p → q**	**¬ (p → q)**	**¬q**	**¬ (p → q) → ¬q**
F	F	T	F	T	T
F	T	T	F	F	T
T	F	F	T	T	T
T	T	T	F	F	T

Here, the last column consists of all true truth values. So, **¬ (p → q) → ¬q** is always true and hence is a tautology.

Contradiction:

A compound proposition that is always false, no matter what the truth values of the propositional variables that occur in it, is called a contradiction.

Example 1:

Show that p ∧ ¬p is a contradiction.

Solution:

We can verify this statement with help of truth table as follows:

Truth Table:

p	**¬p**	**p ∧ ¬p**
T	F	F
F	T	F

Here, the last column consists of all false truth values. So, p ∧ ¬p is always false and hence is a contradiction.

Example 2:

Show that (p ∨ q) ∧ [(¬p) ∧ (¬q)] is contradiction.

Solution:

We can verify this statement with help of truth table as follows:

Truth Table:

p	**q**	**p ∨ q**	**¬ p**	**¬q**	**(¬p) ∧ (¬q)]**	**(p ∨ q) ∧ [(¬p) ∧ (¬q)]**
F	F	F	T	T	T	F
F	T	T	T	F	F	F
T	F	T	F	T	F	F
T	T	T	F	F	F	F

Here, the last column consists of all false truth values. So, **(p ∨ q) ∧ [(¬p) ∧ (¬q)]** is always false and hence is a contradiction.

Contingency:

A compound proposition that is neither a tautology nor a contradiction, no matter what the truth values of the propositional variables that occur in it, is called a contradiction.

Example:

Show that ¬p ∧ ¬q is contingency.

Solution:

We can verify this statement with help of truth table as follows:

Truth Table:

p	**q**	**¬ p**	**¬q**	**¬p ∧ ¬q**
F	F	T	T	T
F	T	T	F	F

T	F	F	T	F
T	T	F	F	F

Here, the last column consists of neither all true nor false truth values. So, **¬p ∧ ¬q** is neither a tautology or a contradiction, and hence is a contingency.

Logical Equivalences:

Given two propositions that differ in their syntax, we may get the exactly same semantic for both the propositions. Compound propositions that have the same truth values in all possible cases are called logically equivalent. The compound propositions p and q are called logically equivalent if p ↔ q is a tautology. The notation p ≡ q denotes that p and q are logically equivalent.

The symbol ≡ is not a logical connective, and p ≡ q is not a compound proposition but rather is the statement that p ↔ q is a tautology. The symbol ⇔ is sometimes used instead of ≡ to denote logical equivalence.

If two propositions are semantically identical, then we say those two propositions are equivalent. To test whether two propositions p and q are logically equivalent, the following steps are followed:

i. Construct the truth table for p and q.
ii. Check each combination of truth values of propositional variables to see whether the value of p is same as value of q or not.
iii. If truth value of p is same as truth value of q, then p and q are logically equivalent.

Example: Prove that (p → r) **∨** (q → r) ≡ (p ∧ q) → r

Solution:

p	q	r	p → r	q → r	**(p → r) ∨ (q → r)**	p ∧ q	**(p ∧ q) → r**
F	F	F	T	T	T	F	T
F	F	T	T	T	T	F	T
F	T	F	T	F	T	F	T
F	T	T	T	T	T	F	T
T	F	F	F	T	T	F	T

T	F	T	T	T	T	F	T
T	T	F	F	F	F	T	F
T	T	T	T	T	T	T	T

Therefore, (p → r) **V** (q → r) ≡ (p ∧ q) → r

TABLE 6 Logical Equivalences.

Equivalence	***Name***
$p \wedge \mathbf{T} \equiv p$ $p \vee \mathbf{F} \equiv p$	Identity laws
$p \vee \mathbf{T} \equiv \mathbf{T}$ $p \wedge \mathbf{F} \equiv \mathbf{F}$	Domination laws
$p \vee p \equiv p$ $p \wedge p \equiv p$	Idempotent laws
$\neg(\neg p) \equiv p$	Double negation law
$p \vee q \equiv q \vee p$ $p \wedge q \equiv q \wedge p$	Commutative laws
$(p \vee q) \vee r \equiv p \vee (q \vee r)$ $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$	Associative laws
$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$ $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$	Distributive laws
$\neg(p \wedge q) \equiv \neg p \vee \neg q$ $\neg(p \vee q) \equiv \neg p \wedge \neg q$	De Morgan's laws
$p \vee (p \wedge q) \equiv p$ $p \wedge (p \vee q) \equiv p$	Absorption laws
$p \vee \neg p \equiv \mathbf{T}$ $p \wedge \neg p \equiv \mathbf{F}$	Negation laws

TABLE 7 Logical Equivalences Involving Conditional Statements.
$p \to q \equiv \neg p \vee q$
$p \to q \equiv \neg q \to \neg p$
$p \vee q \equiv \neg p \to q$
$p \wedge q \equiv \neg(p \to \neg q)$
$\neg(p \to q) \equiv p \wedge \neg q$
$(p \to q) \wedge (p \to r) \equiv p \to (q \wedge r)$
$(p \to r) \wedge (q \to r) \equiv (p \vee q) \to r$
$(p \to q) \vee (p \to r) \equiv p \to (q \vee r)$
$(p \to r) \vee (q \to r) \equiv (p \wedge q) \to r$

TABLE 8 Logical Equivalences Involving Biconditional Statements.
$p \leftrightarrow q \equiv (p \to q) \wedge (q \to p)$
$p \leftrightarrow q \equiv \neg p \leftrightarrow \neg q$
$p \leftrightarrow q \equiv (p \wedge q) \vee (\neg p \wedge \neg q)$
$\neg(p \leftrightarrow q) \equiv p \leftrightarrow \neg q$

Example: Show that p→q and ¬p∨q are logically equivalent.

Solution:

Let's construct the truth table for these compound propositions.

TABLE 4 **Truth Tables for $\neg p \vee q$ and $p \to q$.**

p	q	$\neg p$	$\neg p \vee q$	$p \to q$
T	T	F	T	T
T	F	F	F	F
F	T	T	T	T
F	F	T	T	T

Because the truth values of ¬p ∨ q and p → q agree, they are logically equivalent.

Identity Law:

a. p ∧ T ≡ p

b. p ∨ F ≡ p

Verification:

p	p ∧ T	p ∨ F
T	T	T
F	F	F

Domination Law:

a. p ∨ T ≡ T

b. p ∧ F ≡ F

Verification:

p	p ∨ T	p ∧ F
T	T	F
F	T	F

Idempotent Law:

a. p ∨ p ≡ p

b. p ∧ p ≡ p

Verification:

p	p ∧ p	p ∨ p
T	T	T
F	F	F

Double Negation Law:

¬(¬p) ≡ p
Verification:

p	¬p	¬(¬p)
T	F	T
F	T	F

Commutative Law:

a. p ∨ q ≡ q ∨ p
b. p ∧ q ≡ q ∧ p

Verification:

p	q	p ∨ q	q ∨ p	p ∧ q	q ∧ p
T	T	T	T	T	T
T	F	T	T	F	F
F	T	T	T	F	F
F	F	F	F	F	F

Associative laws:

a. (p ∨ q) ∨ r ≡ p ∨ (q ∨ r)
b. (p ∧ q) ∧ r ≡ p ∧ (q ∧ r)

Verification:

p	q	r	p ∨ q	**(p ∨ q) ∨ r**	q ∨ r	**p ∨ (q ∨ r)**
T	T	T	T	T	T	T
T	T	F	T	T	T	T
T	F	T	T	T	T	T
T	F	F	T	T	F	T
F	T	T	T	T	T	T
F	T	F	T	T	T	T
F	F	T	F	T	T	T
F	F	F	F	F	F	F

p	q	r	p ∧ q	**(p ∧ q) ∧ r**	q ∧ r	**p ∧ (q ∧ r)**
T	T	T	T	T	T	T
T	T	F	T	F	F	F
T	F	T	F	F	F	F
T	F	F	F	F	F	F
F	T	T	F	F	T	F
F	T	F	F	F	F	F
F	F	T	F	F	F	F
F	F	F	F	F	F	F

Distributive laws:

a. p ∨ (q ∧ r) ≡ (p ∨ q) ∧ (p ∨ r)
b. p ∧ (q ∨ r) ≡ (p ∧ q) ∨ (p ∧ r)

Verification:

p	q	r	q ∧ r	**p ∨ (q ∧ r)**	p ∨ q	p ∨ r	**(p ∨ q) ∧ (p ∨ r)**
T	T	T	T	T	T	T	T
T	T	F	F	T	T	T	T
T	F	T	F	T	T	T	T
T	F	F	F	T	T	T	T
F	T	T	T	T	T	T	T
F	T	F	F	F	T	F	F
F	F	T	F	F	F	T	F
F	F	F	F	F	F	F	F

Similarly, p ∧ (q ∨ r) ≡ (p ∧ q) ∨ (p ∧ r) can be verified.

De-Morgan's laws:

a. ¬(p ∧ q) ≡ ¬p ∨ ¬q
b. ¬(p ∨ q) ≡ ¬p ∧ ¬q

Proof:

p	q	$p \vee q$	$\neg(p \vee q)$	$\neg p$	$\neg q$	$\neg p \wedge \neg q$
T	T	T	F	F	F	F
T	F	T	F	F	T	F
F	T	T	F	T	F	F
F	F	F	T	T	T	T

Similarly, ¬ (p ∧ q) ≡ ¬p ∨ ¬q can be proved.

Assignment: Prove Absorption Laws and Negation Laws using truth table.

Predicates and Quantifiers:

Propositional logic cannot adequately express the meaning of all statements in mathematics and in natural language.

For example, suppose that we know that

"Every computer connected to the university network is functioning properly."

No rules of propositional logic allow us to conclude the truth of the statement “MATH3 is functioning properly,” where MATH3 is one of the computers connected to the university network.

Likewise, we cannot use the rules of propositional logic to conclude from the statement

“CS2 is under attack by an intruder,”

where CS2 is a computer on the university network, to conclude the truth of

“There is a computer on the university network that is under attack by an intruder.”

We need a more powerful type of logic called predicate logic, and will see how predicate logic can be used to express the meaning of a wide range of statements in mathematics and computer science in ways that permit us to reason and explore relationships between objects.

Predicates:

Any declarative statements involving variables often found in mathematics and computer programs, which are neither true nor false when the values of variables are not specified, is called predicate.

The logic involving predicates is called Predicate Logic or Predicate Calculus.

For example:

Statements involving variables, such as “$x > 3$,” “$x = y + 3$,” “$x + y = z$,” and “Computer x is under attack by an intruder,” and “Computer x is functioning properly”.

We need to find the ways that propositions can be produced from such statements.

The statement “x is greater than 3” has two parts. The first part, the variable x, is the subject of the statement. The second part—the predicate, “is greater than 3”—refers to a property that the subject of the statement can have. We can denote the statement “x is greater than 3” by $P(x)$, where P denotes the predicate “is greater than 3” and x is the variable. The statement $P(x)$ is also said to be the value of the propositional function P at x. Once a value has been assigned to the variable x, the statement $P(x)$ becomes a proposition and has a truth value.

<u>Example:</u>

Let $P(x)$ denote the statement “$x > 3$.” What are the truth values of $P(4)$ and $P(2)$?

<u>Solution:</u> We obtain the statement $P(4)$ by setting $x = 4$ in the statement “$x > 3$.” Hence, $P(4)$, which is the statement “$4 > 3$,” is true. However, $P(2)$, which is the statement “$2 > 3$,” is false.

We can also have statements that involve more than one variable. For instance, consider the statement "x = y + 3." We can denote this statement by Q(x, y), where x and y are variables and Q is the predicate. When values are assigned to the variables x and y, the statement Q(x, y) has a truth value.

<u>Example:</u>

Let Q(x, y) denote the statement "x = y + 3." What are the truth values of the propositions Q(1, 2) and Q(3, 0)?

<u>Solution:</u> To obtain Q(1, 2), set x = 1 and y = 2 in the statement Q(x, y).

Hence, Q(1, 2) is the statement "1 = 2 + 3," which is false.

The statement Q(3, 0) is the proposition "3 = 0 + 3," which is true.

Assignment: Let R (x, y, z) denote the statement "x + y = z." When values are assigned to the variables x, y, and z, this statement has a truth value. Find the truth values of the propositions R (1, 2, 3) and R (0, 0, 1).

<u>Example:</u>

Let A(c, n) denote the statement "Computer c is connected to network n," where c is a variable representing a computer and n is a variable representing a network. Suppose that the computer MATH1 is connected to network CAMPUS2, but not to network CAMPUS1. What are the values of A(MATH1, CAMPUS1) and A(MATH1, CAMPUS2)?

<u>Solution:</u> Because MATH1 is not connected to the CAMPUS1 network, we see that A(MATH1, CAMPUS1) is false. However, because MATH1 is connected to the CAMPUS2 network, we see that A(MATH1, CAMPUS2) is true.

Quantifiers:

When the variables in a propositional function are assigned values, the resulting statement becomes a proposition with a certain truth value. However, there is another important way, called **quantification**, to create a proposition from a propositional function. Quantification expresses the extent to which a predicate is true over a range of elements. In English, the words all, some, many, none, and few are used in quantifications. We will focus on two types of quantification here: universal quantification, which tells us that a predicate is true for every element under consideration, and existential quantification, which tells us that there is one or more element under consideration for which the predicate is true. The area of logic that deals with predicates and quantifiers is called the **predicate calculus**.

Construction of propositions from the predicates using quantifiers is called the quantification.

The variables that appear in the statement can take different possible values and all the possible values that the variable can take forms a domain called "Universe of Discourse" or "Universal Set".

Universal Quantifier:

The universal quantification of P(x) is the statement

"P(x) for all values of x in the domain."

The notation ∀xP(x) denotes the universal quantification of P(x). Here ∀ is called the universal quantifier. We read ∀xP(x) as "for all xP(x)" or "for every xP(x)." An element for which P(x) is false is called a counterexample to ∀xP(x).

<u>Example:</u>

Take universe of discourse as the set of all students of NCC BIM. P (x) represents: x takes Discrete Structure class.

Here universal quantification is ∀x P(x), which represents the English sentence "all students of NCC BIM take Discrete Structure class", and now is a proposition.

<u>Example:</u>

What does the statement ∀xN(x) mean if N(x) is "Computer x is connected to the network" and the domain consists of all computers on campus?

Solution:

The statement ∀xN(x) means that for every computer x on campus, that computer x is connected to the network. This statement can be expressed in English as "Every computer on campus is connected to the network."

The universal quantification is conjunction of all the propositions that are obtained by assigning value of the variable in the predicate. If universe of discourse is a set {Ram, Shyam, Hari, Sita}, then the truth value of the universal quantification is given by P(ram) ^ P(Shyam) ^ P(Hari) ^ P(Sita). i.e., it is true only if all the atomic propositions are true.

Example:

Let P(x) be the statement "x + 1 > x." What is the truth value of the quantification ∀xP(x), where the domain consists of all real numbers?

Solution: Because P(x) is true for all real numbers x, the quantification ∀xP(x) is true.

Example:

Let Q(x) be the statement "x < 2." What is the truth value of the quantification ∀xQ(x), where the domain consists of all real numbers?

Solution: Q(x) is not true for every real number x, because, for instance, Q(3) is false. That is, x = 3 is a counterexample for the statement ∀xQ(x). Thus, ∀xQ(x) is false.

Example:

What is the truth value of ∀xP(x) where P(x) is the statement "x^2<10" and the domain consists of the positive integers not exceeding 4?

Solution:

The statement ∀xP(x) is same as the conjunction P(1)^P(2)^P(3)^P(4) since the domain consists of the integers 1,2,3,4.

Because P(4) is the statement "4^2<10", which is "16<10", is false, it follows that ∀xP(x) is false.

Existential Quantifier:

The existential quantification of P(x) is the proposition

"There exists an element x in the domain such that P(x)."

We use the notation ∃xP(x) for the existential quantification of P(x). Here ∃ is called the existential quantifier.

With existential quantification, we form a proposition that is true if and only if P(x) is true for at least one value of x in the domain.

Besides the phrase "there exists," we can also express existential quantification in many other ways, such as by using the words "for some," "for at least one," or "there is."

The existential quantification ∃xP(x) is read as "There is an x such that P(x)," "There is at least one x such that P(x)," or "For some xP(x)."

Generally, an implicit assumption is made that all domains of discourse for quantifiers are nonempty. If the domain is empty, then ∃xP(x) is false whenever P(x) is a propositional function because when the domain is empty, there can be no element x in the domain for which P(x) is true.

The existential quantification is disjunction of all the propositions that are obtained by assigning value of the variable in the predicate. If universe of discourse is a set {Ram, Shyam, Hari, Sita}, then the truth value of the existential quantification is given by P(ram) ∨ P(Shyam) ∨ P(Hari) ∨ P(Sita). i.e., it is true when one of the atomic propositions are true.

TABLE 1 Quantifiers.

Statement	*When True?*	*When False?*
$\forall x P(x)$	$P(x)$ is true for every x.	There is an x for which $P(x)$ is false.
$\exists x P(x)$	There is an x for which $P(x)$ is true.	$P(x)$ is false for every x.

Example:

Take universe of discourse as the set of all students of NCC BIM. P (x) represents: x takes Discrete Structure class.

Here existential quantification is ∃xP(x), which represents the English sentence "some students of NCC BIM take Discrete Structure class" or "at least one of the students of NCC BIM take Discrete Structure class", and now is a proposition.

Example:

What is the truth value of ∃xP(x) where P(x) is the statement "x^2<10" and the domain consists of the positive integers not exceeding 4?

Solution:

The statement ∃xP(x) is same as the conjunction P(1) ∨ P(2) ∨ P(3) ∨ P(4) since the domain consists of the integers 1,2,3,4.

Because P(1), P(2) and P(3) satisfy "x^2<10", it follows that ∃xP(x) is true.

Example:

Let P(x) denote the statement "$x > 3$." What is the truth value of the quantification $\exists x P(x)$, where the domain consists of all real numbers?

Solution:

Because "$x > 3$" is sometimes true—for instance, when $x = 4$—the existential quantification of P(x), which is $\exists x P(x)$, is true.

Example:

Let Q(x) denote the statement "$x = x + 1$." What is the truth value of the quantification $\exists x Q(x)$, where the domain consists of all real numbers?

Solution:

Because Q(x) is false for every real number x, the existential quantification of Q(x), which is $\exists x Q(x)$, is false.

Example:

Let Z, the set of integers, be the universe of domain and consider the statements:

a. $\forall x \in Z, x^2=x$
b. $\exists x \in Z, x^2=x$

Find the truth values of each of the statements.

Solution:

Let P(x): $x^2=x$ then

$\forall x P(x)$ is false because $2 \in Z$ and P(2): $2^2=2$ is false.

$\exists x P(x)$ is true because $1 \in Z$ and P(1): $1^2=1$ is true.

Example:

Let N= {1,2,3,4,5,6,7,8,9} be set of natural numbers. Determine the truth value of each of the following statements.

a. ∃x ∈ N, x+5 = 12

b. ∀x ∈ N, x+4 < 15

c. ∀x ∈ N, x+5 <= 10

Solution:

Given,

N= {1,2,3,4,5,6,7,8,9}

a. ∃x ∈ N, x+5 = 12 is true. For x=7, 7+5=12 is true.

b. ∀x ∈ N, x+4 < 15 is true. For every x ∈ N, x+4 < 15 satisfies the condition.

c. ∀x ∈ N, x+5 <= 10 is false. For 6 ∈ N, 6+5 <= 10 is false.

Precedence of Quantifiers:

The quantifiers ∀ and ∃ have higher precedence than all logical operators from propositional calculus. For example, ∀xP(x) ∨ Q(x) is the disjunction of ∀xP(x) and Q(x). In other words, it means (∀xP(x)) ∨ Q(x) rather than ∀x(P(x) ∨ Q(x)).

Binding Variables:

When a quantifier is used on the variable x, we say that this occurrence of the variable is bound. An occurrence of a variable that is not bound by a quantifier or set equal to a particular value is said to be free. All the variables that occur in a propositional function must be bound or set equal to a particular value to turn it into a proposition.

Examples:

- P (x, y) has two free variables x and y.
- P (2, y) has one bound variable 2 and one free variable y.
- P (2, y) where y=4, is bounding the variable y also.
- ∀x P(x) has a bound variable x.
- ∀x P(x, y) has one bound variable x and one free variable y.

Note: Expression with no free variables is a proposition. Expression with at least one free variable is a predicate only.

Example:

In the statement ∀y P (x, y), the variable y is bounded by universal quantifier ∀y but the variable x is free because it is not bound by a quantifier.

Negation of Quantified Expression:

Let P (x) denotes x is hardworking and universe of discourse for x is: Students in NCC. Then, ∀x P(x) is every student in NCC is hardworking. If we want to negate it, the meaning would be like: There is a student in NCC who is not hardworking. i.e., ∃x ¬ P(x).

∃x P(x) is at least a student in NCC is hardworking. The negation would be no student in NCC is hardworking. i.e., ∀x ¬ P(x).

Example:

Find the negation of the statement "Every student in your class has taken a course in calculus."

Solution:

This statement is a universal quantification, namely, ∀xP(x), where P(x) is the statement "x has taken a course in calculus" and the domain consists of the students in your class. The negation of this statement is "It is not the case that every student in your class has taken a course in calculus." This is equivalent to "There is a student in your class who has not taken a course in calculus." And this is simply the existential quantification of the negation of the original propositional function, namely, ∃x ¬P(x).

This above example illustrates the following logical equivalence: ¬∀xP(x) ≡ ∃x ¬P(x).

To show that ¬∀xP(x) and ∃x ¬P(x) are logically equivalent no matter what the propositional function P(x) is and what the domain is, first note that ¬∀xP(x) is true if and only if ∀xP(x) is false. Next, note that ∀xP(x) is false if and only if there is an element x in the domain for which P(x) is false. This holds if and only if there is an element x in the domain for which ¬P(x) is true. Finally, note that there is an element x in the domain for which ¬P(x) is true if and only if ∃x ¬P(x) is true. Putting these steps together, we can conclude that ¬∀xP(x) is true if and only if ∃x ¬P(x) is true. It follows that ¬∀xP(x) and ∃x ¬P(x) are logically equivalent.

Example:

What are the negations of the statements “There is an honest politician” and “All Nepalis eat dalbhat”?

Solution:

Let P(x) denote “x is honest.” Then the statement “There is an honest politician” is represented by $\exists x\ P(x)$, where the domain consists of all politicians. The negation of this statement is $\neg\exists x\ P(x)$, which is equivalent to $\forall x\ \neg P(x)$. This negation can be expressed as “Every politician is dishonest.” (Note: In English, the statement “All politicians are not honest” is ambiguous. In common usage, this statement often means “Not all politicians are honest.” Consequently, we do not use this statement to express this negation.)

Extra Let Q(x) denote “x eats dalbhat.” Then the statement “All Nepalis eat dalbhat” is represented by $\forall x\ Q(x)$, where the domain consists of all Nepalis. The negation of this statement is $\neg\forall x\ Q(x)$, which is equivalent to $\exists x\ \neg Q(x)$. This negation can be expressed in several different ways, including “Some Nepalis does not eat dalbhat” and “There is a Nepali who does not eat dalbhat.”

Translating Sentences into Logical Expressions using Predicates and Quantifiers:

Example:

Translate “not every integer is even” where the universe of discourse is set of integers.

Solution:

Let E (x) denote x is even.

Then, $\neg\forall x\ E(x)$ represents the statement “not every integer is even”.

<u>Example:</u>

Express the statement “Every student in this class has studied calculus” using predicates and quantifiers.

Solution:

First, we rewrite the statement so that we can clearly identify the appropriate quantifiers to use. Doing so, we obtain: “For every student in this class, that student has studied calculus.” Next, we introduce a variable x so that our statement becomes “For every student x in this class, x has studied calculus." Continuing, we introduce C(x), which is the statement "x has studied calculus." Consequently, if the domain for x consists of the students in the class, we can translate our statement as $\forall x\, C(x)$.

If we change the domain to consist of all people, we will need to express our statement as

"For every person x, if person x is a student in this class, then x has studied calculus."

If S(x) represents the statement that person x is in this class, we see that our statement can be expressed as $\forall x\, (S(x) \rightarrow C(x))$.

<u>Example:</u>

Express the statements "Some student in this class has visited Mexico" and "Every student in this class has visited either Canada or Mexico" using predicates and quantifiers.

Solution:

The statement "Some student in this class has visited Mexico" means that "There is a student in this class with the property that the student has visited Mexico."

We can introduce a variable x, so that our statement becomes "There is a student x in this class having the property that x has visited Mexico." We introduce M(x), which is the statement "x has visited Mexico." If the domain for x consists of the students in this class, we can translate this first statement as $\exists x\, M(x)$.

However, if we are interested in people other than those in this class, we look at the statement a little differently. Our statement can be expressed as "There is a person x having the properties that x is a student in this class and x has visited Mexico." In this case, the domain for the variable x consists of all people. We introduce S(x) to represent "x is a student in this class." Our solution becomes $\exists x(S(x) \wedge M(x))$ because the statement is that there is a person x who is a student in this class and who has visited Mexico.

Similarly, the second statement can be expressed as:

"For every x in this class, x has the property that x has visited Mexico or x has visited Canada."

We let C(x) be "x has visited Canada." We see that if the domain for x consists of the students in this class, this second statement can be expressed as $\forall x(C(x) \vee M(x))$.

However, if the domain for x consists of all people, our statement can be expressed as

"For every person x, if x is a student in this class, then x has visited Mexico or x has visited Canada."

In this case, the statement can be expressed as $\forall x\ (S(x) \rightarrow (C(x) \lor M(x)))$.

Example: Express the following sentence using quantifier.

Every person is precious.

Solution:

We translate this as:

For every x, if x is a person, then x is precious.

P(x): x is a person.

Q(x): x is precious.

So, the translation will be: $\forall x, P(x) \rightarrow Q(x)$

Example: Express the following sentence using quantifier.

Some student of this college passed CSIT entrance examination.

Solution:

The translation will be:

For some x, x is a student of this college and x has passed CSIT entrance examination.

Let C(x): x is a student of this college.

E(x): x passed CSIT entrance examination.

Then the translation will be: $\exists x, C(x) \land E(x)$

Where universe of discourse is set of all students.

Example:

Let S(x): x is a student.

C(x): x is clever.

M(x): x is successful.

Express the following sentences using quantifier.

a. There exists a student.
b. Some students are clever.
c. Some students are not successful.

Solution:
a. There exists a student: $\exists x\, S(x)$
b. Some students are clever. This can be written as: There exists an x such that x is a student and x is clever: $\exists x\, (S(x) \wedge C(x))$
c. Some students are not successful. This can be expressed as: There exists an x such that x is student and x is not successful: $\exists x\, (S(x) \wedge \neg M(x))$

Nested Quantifiers:

When we use more than one quantifier in a sequence, then it is known as nested quantifier. For example: $\forall x\, \exists y\, P(x, y)$ where $P(x, y)$: $x + y = 0$. Here, quantifiers $\forall$ and $\exists$ are used in sequence.

Example:

Translate the following sentences using quantifiers:

a. Everyone loves someone.
b. Someone loves somebody.
c. Everyone loves everybody.

Solution:

Let $L(x, y)$: x loves y.

Then,

a. For all x, there is some y such that x loves y.

   $\forall x\, \exists y\, L(x, y)$

b. For some x, there is some y such that x loves y.

   $\exists x\, \exists y\, L(x, y)$

c. For all x and for all y, x loves y.

   $\forall x\, \forall y\, L(x, y)$

Example:

Translate the statement “The sum of two positive integers is always positive” into a logical expression.

Solution:

To translate this statement into a logical expression, we first rewrite it so that the implied quantifiers and a domain are shown: “For every two integers, if these integers are both positive, then the sum of these integers is positive.”

Next, we introduce the variables x and y to obtain “For all positive integers x and y, x + y is positive.”

Consequently, we can express this statement as $\forall x \forall y((x > 0) \land (y > 0) \rightarrow (x + y > 0))$, where the domain for both variables consists of all integers.

Example:

Translate the statement

$$\forall x(C(x) \lor \exists y(C(y) \land F(x, y)))$$

into English, where C(x) is “x has a computer,” F(x, y) is “x and y are friends,” and the domain for both x and y consists of all students in your school.

Solution:

The statement says that for every student x in your school, x has a computer or there is a student y such that y has a computer and x and y are friends.

In other words, every student in your school has a computer or has a friend who has a computer.

Example:

Express the statement “If a person is female and is a parent, then this person is someone’s mother” as a logical expression involving predicates, quantifiers with a domain consisting of all people, and logical connectives.

Solution:

The statement “If a person is female and is a parent, then this person is someone’s mother” can be expressed as:

"For every person x, if person x is female and person x is a parent, then there exists a person y such that person x is the mother of person y."

We introduce the propositional functions F(x) to represent "x is female," P(x) to represent "x is a parent," and M (x, y) to represent "x is the mother of y."

The original statement can be represented as $\forall x\ ((F(x) \land P(x)) \rightarrow \exists y\ M(x, y))$.

Rules of Inference:

Proofs in mathematics are valid arguments that establish the truth of mathematical statements. By an argument, we mean a sequence of statements that end with a conclusion. By valid, we mean that the conclusion, or final statement of the argument, must follow from the truth of the preceding statements, or premises, of the argument.

To deduce new statements from statements we already have, we use rules of inference which are templates for constructing valid arguments. Rules of inference are our basic tools for establishing the truth of statements.

An argument in propositional logic is a sequence of propositions. All but the final proposition in the argument are called premises and the final proposition is called the conclusion. An argument is valid if the truth of all its premises implies that the conclusion is true.

Rules of Inference for Propositional Logic:

We can always use a truth table to show that an argument form is valid. We do this by showing that whenever the premises are true, the conclusion must also be true. However, this can be a tedious approach. For example, when an argument form involves 10 different propositional variables, to use a truth table to show this argument form is valid requires $2^{10} = 1024$ different rows. Fortunately, we do not have to resort to truth tables. Instead, we can first establish the validity of some relatively simple argument forms, called rules of inference. These rules of inference can be used as building blocks to construct more complicated valid argument forms. We will now introduce the most important rules of inference in propositional logic.

TABLE 1 Rules of Inference.		
Rule of Inference	***Tautology***	***Name***
p $p \to q$ $\therefore q$	$(p \land (p \to q)) \to q$	Modus ponens
$\neg q$ $p \to q$ $\therefore \neg p$	$(\neg q \land (p \to q)) \to \neg p$	Modus tollens
$p \to q$ $q \to r$ $\therefore p \to r$	$((p \to q) \land (q \to r)) \to (p \to r)$	Hypothetical syllogism
$p \lor q$ $\neg p$ $\therefore q$	$((p \lor q) \land \neg p) \to q$	Disjunctive syllogism
p $\therefore p \lor q$	$p \to (p \lor q)$	Addition
$p \land q$ $\therefore p$	$(p \land q) \to p$	Simplification
p q $\therefore p \land q$	$((p) \land (q)) \to (p \land q)$	Conjunction
$p \lor q$ $\neg p \lor r$ $\therefore q \lor r$	$((p \lor q) \land (\neg p \lor r)) \to (q \lor r)$	Resolution

Example 1:

Consider the following argument involving propositions (which, by definition, is a sequence of propositions):

"If you have a current password, then you can log onto the network."

"You have a current password."

Therefore, “You can log onto the network.”

We would like to determine whether this is a valid argument. That is, we would like to determine whether the conclusion “You can log onto the network” must be true when the premises “If you have a current password, then you can log onto the network” and “You have a current password” are both true.

Use p to represent “You have a current password” and q to represent “You can log onto the network.” Then, the argument has the form:

$$\begin{array}{rl} & p \\ & p \to q \\ \hline \therefore & q \end{array}$$

Example 2: Prove the following statement using Modus Ponens rule.

“If you have access to the network, then you can change your grade.”
“You have access to the network.”

∴ “You can change your grade.”

Example 3:

“If it snows today, then we will go skiing”,

“It is snowing today,

Find the conclusion of the conditional statement.

Example 4:

Given that: If Ram is human, then Ram is mortal. Ram is human. Prove that Ram is mortal by using rules of inference.

Example 5:

State which rule of inference is the basis of the following argument: “It is below freezing and raining now. Therefore, it is below freezing now.”

Solution: Let p be the proposition “It is below freezing now,” and let q be the proposition “It is raining now.” This argument is of the form

$$\begin{array}{rl} & p \wedge q \\ \hline \therefore & p \end{array}$$

This argument uses the simplification rule. ◀

Example 6:

State which rule of inference is used in the argument:

If it rains today, then we will not have a barbecue today. If we do not have a barbecue today, then we will have a barbecue tomorrow. Therefore, if it rains today, then we will have a barbecue tomorrow.

Solution: Let p be the proposition "It is raining today," let q be the proposition "We will not have a barbecue today," and let r be the proposition "We will have a barbecue tomorrow." Then this argument is of the form

$$\begin{array}{rl} & p \to q \\ & q \to r \\ \hline \therefore & p \to r \end{array}$$

Hence, this argument is a hypothetical syllogism. ◀

Example 7:

Show that the premises "If you send me an e-mail message, then I will finish writing the program," "If you do not send me an e-mail message, then I will go to sleep early," and "If I go to sleep early, then I will wake up feeling refreshed" lead to the conclusion "If I do not finish writing the program, then I will wake up feeling refreshed."

Solution:

Let p be the proposition "You send me an e-mail message,"

q the proposition "I will finish writing the program,"

r the proposition "I will go to sleep early," and

s the proposition "I will wake up feeling refreshed."

Then the premises are $p \to q$, $\neg p \to r$, and $r \to s$. The desired conclusion is $\neg q \to s$.

We need to give a valid argument with premises $p \to q$, $\neg p \to r$, and $r \to s$ and conclusion $\neg q \to s$.

This argument form shows that the premises lead to the desired conclusion.

Step	Reason
1. $p \to q$	Premise
2. $\neg q \to \neg p$	Contrapositive of (1)
3. $\neg p \to r$	Premise
4. $\neg q \to r$	Hypothetical syllogism using (2) and (3)
5. $r \to s$	Premise
6. $\neg q \to s$	Hypothetical syllogism using (4) and (5)

Example 8:

Show that the premises "It is not sunny this afternoon and it is colder than yesterday," "We will go swimming only if it is sunny," "If we do not go swimming, then we will take a canoe trip," and "If we take a canoe trip, then we will be home by sunset" lead to the conclusion "We will be home by sunset."

Solution: Let p be the proposition "It is sunny this afternoon," q the proposition "It is colder than yesterday," r the proposition "We will go swimming," s the proposition "We will take a canoe trip," and t the proposition "We will be home by sunset." Then the premises become $\neg p \wedge q$, $r \rightarrow p$, $\neg r \rightarrow s$, and $s \rightarrow t$. The conclusion is simply t. We need to give a valid argument with premises $\neg p \wedge q$, $r \rightarrow p$, $\neg r \rightarrow s$, and $s \rightarrow t$ and conclusion t.

We construct an argument to show that our premises lead to the desired conclusion as follows.

Step	Reason
1. $\neg p \wedge q$	Premise
2. $\neg p$	Simplification using (1)
3. $r \rightarrow p$	Premise
4. $\neg r$	Modus tollens using (2) and (3)
5. $\neg r \rightarrow s$	Premise
6. s	Modus ponens using (4) and (5)
7. $s \rightarrow t$	Premise
8. t	Modus ponens using (6) and (7)

Note that we could have used a truth table to show that whenever each of the four hypotheses is true, the conclusion is also true. However, because we are working with five propositional variables, p, q, r, s, and t, such a truth table would have 32 rows. ◀