1 Propositional logic and Predicates

Propositional Logic

Introduction:

It has a large amount of applications in mathematics. It is used to prove theorems. Every theorem consists of various statements which ultimately reach a conclusion. To check the validation of these statements, logic is required. Mathematical logic tells us whether the statement is valid or not. Now, the question that may arise is, from where do all these statements come from? So, all these statements come from a set of all English statements.

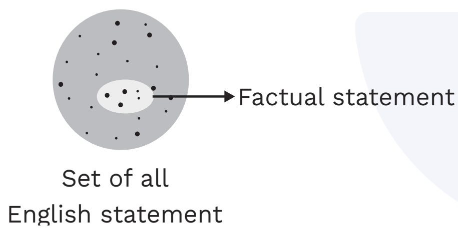


Fig. 1.1

Note:

- Factual statements are part of the "set of all English statements" which is used to conclude something. These factual statements in mathematical logic are called proportional logic.
- Factual statements are facts which have only two cases yes(true)/no(false)
- Any **question, command exclamation, vague reference** will never be a proposition.

1. Which of the following is/are not proposition?
   (A) $x + 2 = 5$
   (B) He is tall
   (C) Today is Monday
   (D) Tomorrow will be rain

Solution: (A), (B), (C), (D)

Definition

Proposition: Proposition is a logical statement which can be true or false but not both.
e.g: 1. $2 + 2 = 4$
2. 'C' is a vowel

Rack Your Brain

Is "*Read this carefully*" a proposition?

Atomic and compound proposition:

The Propositional statement is of two types

1. Compound propositional statement
2. Atomic propositional statement

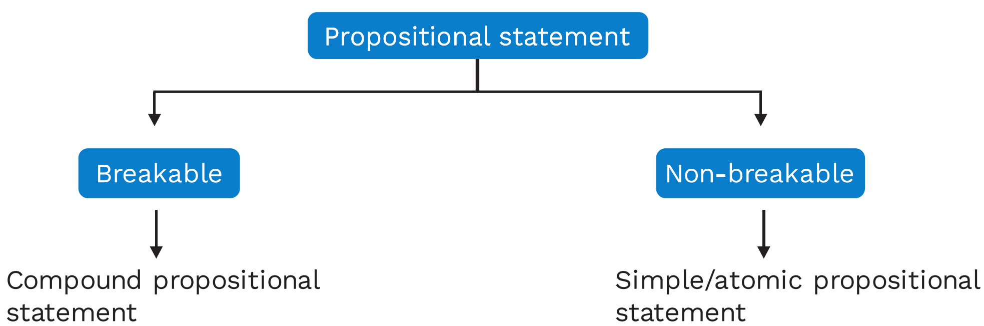


Fig. 1.2

Statements that are constructed by combining one or more factual statements, to form a new proposition is called a compound proposition.

e.g: 1. 5 > 3: non-breakable, ∴ atomic

2. 5 ≥ 3: This statement can be broken into
a) 5 > 3
b) 5 = 3

∴Compound statement

Propositional variable:

A variable is used to represent the proposition.

Simple proposition represented by p, q, r...

Compound proposition represented by P, Q, R...

Single propositions have 2 possible truth combinations.

p
T
F

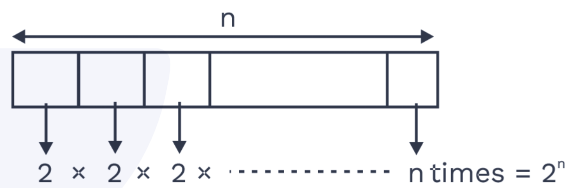
Two propositional variables have $2^2 = 4$ truth combinations.

f	g
0	1
1	0
1	1
0	0

3 propositions variables - 8 truth combinations

⋮

n propositions variables-2^n truth combination



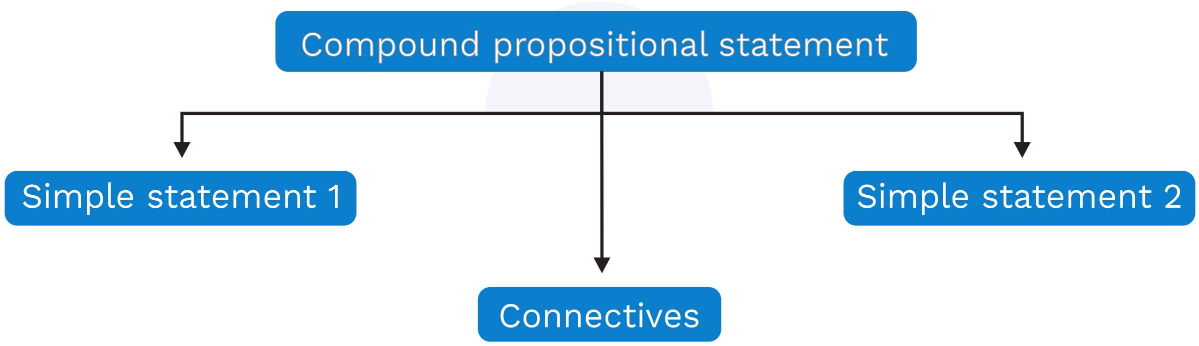


Fig. 1.3

A compound statement is breakable into simple statements while breaking it; it is breakable at the breaking point / weak point called connectives. That weak point is called connectives.

In mathematics, there are 4 types of connectives:

	Representation
• Conjunction / AND	(∧)
• Disjunction / OR	(v)
• Implication	(→/⇒)
• Bi implication	(⇔ /↔)

Note:
There is 1 more type of connective called “modifiers (negation)”.

Boolean logic:

1. Negation:

It is a unary operation and can be denoted in the following ways:

a) $\neg p$
b) $\sim p$
c) p'
d) $\bar{p}$

Note:

A unary operation is an operation which can be implemented on single proposition.

Definition

Negation: Let p be a proposition. The "negation of p", denoted by $\neg$ p (also by $\bar{p}$),

is the statement "It is not the case that p". The proposition $\bar{p}$ is read as "not p".

The truth value of $\bar{p}$ is the opposite of the truth value of p.

e.g: Let a proposition p = I am Michel.
The negation of p($\bar{p}$) = I am not Michel.

p	$\bar{p}$
T	F
F	T

Table 1 The Truth Table for the Negation of a Proposition

2. Conjunction:

It is basically AND operator and is represented by "$\wedge$".

Definition

Conjunction: Conjunction is similar to performing AND operation between two variables. Conjunction between two proposition will be true only when both the propositions are true otherwise it will always be false.

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

Table 2 The Truth Table for Conjunction of Proposition

Note:

In the truth table, we mention the cases. As we all know, the compound statement is nothing but a group of some atomic/simple statements. We generate cases out of these atomic statements to reach the conclusion of the compound statement.

Rack Your Brain

Find the conjunction of the proposition p & q where p is the proposition "It is raining today", and q is the proposition "I will not go to the school".

3. Disjunction:

Disjunction is an "OR" case.
OR is of two types:

a) Inclusive OR ($\vee$)
b) Exclusive OR ($\oplus$)

Note:

In mathematics, we always use (v) inclusive OR, until and unless it is mentioned as $\oplus$

Inclusive OR:

If any of the simple propositions are true, the conclusion will be true and the conclusion will be false otherwise.

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

Table 3 Table for Disjunction Using Inclusive OR

For example,

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

With this statement, we mean that those students who have taken either calculus or computer science can take the class also, the students who have taken both can take the class.

Exclusive OR:

Using exclusive OR, we can reframe the sentence as

"Students who have taken calculus or computer science, but not both can take the class".

q	q	$p \oplus q$
T	T	F
T	F	T
F	T	T
F	F	F

Table 4 Table for Disjunction Using Exclusive OR

Conditional statement/implication:

Definition

Let f and g be two propositions, and the conditional statement $f \rightarrow g$ is the proposition, "if f then g". The conditional statement fg is false when f is true and g is false, and true otherwise.

$f \rightarrow g$

hypothesis/ antecedent/ premisea

conclusion/ consequence

p	q	$p \rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

Table 5 The Truth Table for the Conditional Statement $p \rightarrow q$

Differents form of Implication ($\rightarrow$):

$p \rightarrow q$ can be written as:

- p implies q
- if p then q
- if p, q
- q if p
- q when p
- q whenever p
- q unless p
- p only if q
- a sufficient condition for q is p
- q whenever p
- q is necessary for p
- q follows from p

Rack Your Brain

Convert the given statement into propositional logic form

p: I stay

q: you go

1. I stay if you go
2. I stay only if you go
3. I stay unless you go
4. you go when I stay

1. If you win, I will give you pizza.

Solution:

The statement can be interpreted as: if you win the match, then I will give you pizza.

Case 1: Let p be the proposition " If you win" (hypotheses) and q be conclusion "I will give you pizza".

Case 1 says p and q both are true, means "If you win, I will give you pizza" which is true.

$\therefore$ $p \to q$ = True.

Case 2: p = True = If you win

q = False = I will not give you a pizza

which means, If you win, I will not give you pizza, which is false.

$\therefore p \to q$ = False.

Case 3: Similarly, In case 3 the conclusion is "If you do not win, I will give you pizza" is True. As this condition can not be interpreted from the given statement.

$\therefore$ $p \to q$ = True

Case 4: Thus case says, "If you do not win, I will not give you pizza" Which is true

$\therefore$ $p \to q$ = True.

Rack Your Brain

Let p be the statement "Maria learns discrete mathematics" and q be the statement "Maria will find a good job". Express the statement $p \to q$ as a statement in English.

Converse, contrapositive, and inverse:

These are different types of conditional statements.

Let us consider an implication $p \to q$

- **Contrapositive** $\sim q \to \sim p$
- **Converse** $q \to p$
- **Inverse** $\sim p \to \sim q$

Law of contrapositive:

Implication and its contrapositive are equivalent

$$p \to q \equiv \sim q \to \sim p$$

Converse and Inverse are equivalent.

$$q \to p \equiv \sim p \to \sim q$$

Rack Your Brain

What is the inverse of "The home team wins whenever it is raining".

Bi-conditionals:

It is represented by ($\leftrightarrow$)

Definition

Let p and q be propositions. The biconditional statement $p \leftrightarrow q$, is the proposition "p if and only if q" or "p if q". Biconditional statement between two propositions will be true only when both the propositions have same values(True or false) in all the other cases it will give false output.

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

Table 6 Truth Table for the Biconditional $p \leftrightarrow q$.

e.g., Let p be the statement "you can take the flight" and q be the statement "you buy a ticket". $p \leftrightarrow q$ will be, "you can take the flight if and only if you buy a ticket".

Precedence of logical operator:

Generally, we use parenthesis to define the precedence of a logical operator.

- Negation operator has the highest priority. For example, let us consider a proposition $\neg p \wedge q$. This proposition will be considered as the conjunction of $(\neg p)$ and q and not the negation of the conjunction of p and q.
- Conjunction operator's precedence is greater than the disjunction operator's precedence.
- The conditional and biconditional operator has the lowest precedence among all operators.

Operator	Precedence
$\neg$	1
$\wedge$ $\vee$	2 3
$\rightarrow$ $\leftrightarrow$	4 5

Table 7 Precedence of Logical Operators

Compound propositions:

We have already studied connectives, i.e., conjunction, disjunction, conditional, bi-conditional statements and negations. Now, we can use all these connectives to forms compound propositions with 'n' number of variables.

Construct a truth table for: $(p \vee \neg q) \rightarrow (p \wedge q)$

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

Table 8 Example for Truth Table for Compound Propositions

Rack Your Brain

How many rows appear in a truth table for the following compound statements

(A) $p \rightarrow \neg p$

(B) $(p \vee \neg r) \wedge (q \vee \neg r)$

Generally, we use 1 to represent true and 0 to represent false.

A Boolean variable is a variable that has a value of either 0 or 1.

p	q	$p \vee q$	$p \wedge q$	$p \oplus q$
0	0	0	0	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	0

Table 9 Table for Bit Operators: OR, AND and XOR

Rack Your Brain

Find the bitwise OR, bitwise AND and bitwise XOR for the following pair of strings 11110000, 10101010

Application of propositional logic:

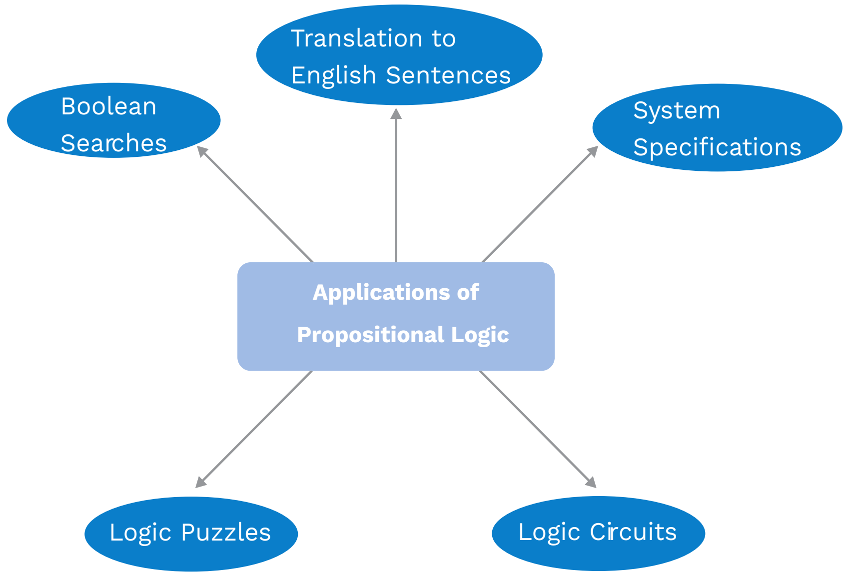


Fig. 1.4

Mathematical logic has a vast number of applications. Some of them we have already mentioned. But we will limit our discussion to only two applications i.e.

(i) Translating English sentences
(ii) Logic circuits

- **Translating English sentences:**
  English is very ambiguous; to resolve that ambiguity, we translate these sentences into compound statements.
- **Example:**
  Convert given sentence into a logical form:
  "You can not ride the roller coaster if you are under 4 feet tall unless you are older than 16 years old."
- **Solution:**
  Let p, q, r represent "You can ride the roller coaster," "you are under 4 feet tall"; and "you are older than 16 years old".
  ∴ The sentence can be translated to
  $(q \wedge \neg r) \rightarrow p$
- **Logic circuits:** Logic is also used to design hardware, which takes the input signal and produces output signals. As we use connectives negation etc., to form compound statements, similarly, in the hardware, we use basic circuits called gates to form the combinational circuit.

(A) AND gate: It takes two signals as input and produces $(p \wedge q)$ signal as output.

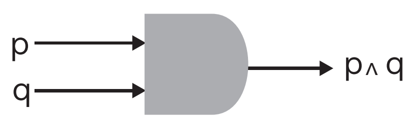


Fig. 1.5

(B) Or gate: It takes two signals as input and gives $(p \vee q)$ signal as output.

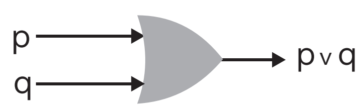


Fig. 1.6

(C) Not gate/inverters:
It takes p as input and produces $(\bar{p})$ as output.

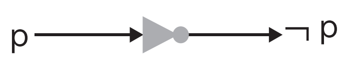


Fig. 1.7

- **Combinatorial circuit:**

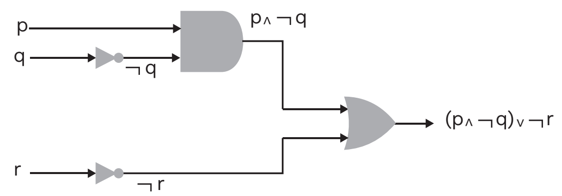


Fig. 1.8

Logically equivalent:

- Two propositions are said to be logically equivalent if and only if they have same truth table.
- Two propositions are said to be logically equivalent if $p \leftrightarrow Q$ is a tautology.
- The Equivalency of two propositions is represented by (≡)

Example 1: Show that $p \rightarrow q$ and $\neg p \vee q$ are logically equivalent.

p	q	$p \rightarrow q$	$\neg p \vee q$
0	0	1	1
0	1	1	1
1	0	0	0
1	1	1	1

Table 1.10

Now, $p \rightarrow q$ and $\neg p \vee q$ are precisely same.

∴ Both boolean expressions are logically equivalent.

Tautology:

If a given proposition turns out to be always true, then it is called a tautology.

Example:

$p \vee \sim p$

p	~q	$p \vee \sim p$
T	F	F
F	T	T

Contradiction:

If a given proposition turns out to be always false, then that proposition is a contradiction.

Example:

$p \wedge \sim p$

p	~p	$p \wedge \sim p$
T	F	F
F	T	T

Contingency:

Given proposition turns out to be either true or false, then that proposition is contingency.

Example:

$(p \wedge q) \rightarrow s$

p	q	s	$p \wedge p \sim s$
T	T	T	F
T	T	F	T

Note:

Contradiction is also known as fallacy or invalid.

Satisfiable:

A proposition that is either a tautology or a contingency is called satisfiable.

Example:

$(p \wedge q \wedge r) \rightarrow s$

q	r	s	$(p \wedge q \wedge \rho) \rightarrow s$
T	T	T	F
T	T	F	T

Contingency so satisfiable

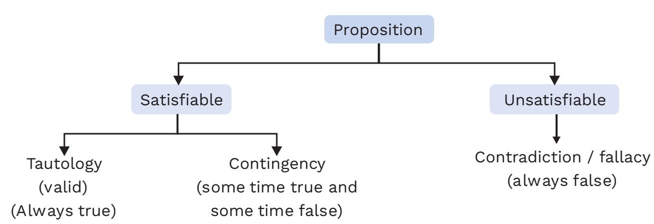


Fig. 1.9

Rack Your Brain

Choose the correct option:

1. All valids are satisfiable
2. All contingency are satisfiable
3. All satisfiable are contingency
4. None

2. Which of the following option/s are satisfiable compound propositions?
(A) $(p \vee \neg q) \wedge (\neg p \quad q) \quad (\neg p \; \neg q)$
(B) $(p \rightarrow q) \wedge (p \quad \neg q) \quad (\neg p \rightarrow q) \; (\neg p \rightarrow \neg q)$
(C) $(p \leftrightarrow q) \wedge (\neg p \leftrightarrow q)$
(D) None of the above

Solution:

(A) $= (p \vee \neg q) \wedge (\neg p \vee q) \wedge (\neg p \vee \neg q)$
$= (p + \neg q)\,(\neg p + q)\,(\neg p + \neg q)$
$= (p \,.\, \neg p + p \,.\, q + \neg p \,.\, \neg q)\,(\neg p + \neg q)$
$= (p \,.\, q + \neg p \,.\, \neg q)\,(\neg p + \neg q)$
$= (\neg p \,.\, \neg q + \neg p \,.\, \neg q)$
$= (\neg p \,.\, \neg q)$

As given preposition depend on the value of ‘p’ and ‘q’ thus it can be true or false therefore satisfiable.

(B) $= (p \rightarrow q) \wedge (p \rightarrow \neg q) \wedge (\neg p \rightarrow q) \wedge (\neg p \rightarrow \neg q)$
$= (\neg p + q)\,(\neg p + \neg q)\,(p + q)\,(p + \neg q)$
$= ((p + q)\,(\neg p + \neg q))\,((p + \neg q)\,(\neg p + q))$
$= 0$

Given preposition is contradiction thus not satisfiable.

(C) $= (p \leftrightarrow q) \wedge (\neg p \leftrightarrow q)$
$= (p \;\text{ss}\!\rightarrow q) \wedge (q \rightarrow p) \wedge (\neg p \rightarrow q) \wedge (q \rightarrow \neg p)$
$= ((\neg p + q)\,(\neg q + p))\,((p + q)\,(\neg q + \neg p))$
$= 0$

It is also not satisfiable.

3. The predicate statement $\forall z[p(z) \rightarrow (\neg q(z) \rightarrow p(z))]$ is:
(A) Satisfiable
(B) Tautology
(C) Contradiction
(D) None of these

Solution:

$\forall z[p(z) \rightarrow (\neg q(z) \rightarrow p(z))]$

$\equiv \neg z[\neg p(z) \vee q(z) \vee p(z)]$

$\equiv \neg \forall z[T \vee q(z)]$

$\equiv \neg T$

$\equiv F$

Previous Years’ Questions

Choose the correct choice regarding the following propositional logic assertion S:
[GATE CSE 2021 Set-2]

S: $((P \wedge Q) \rightarrow R) \rightarrow ((P \wedge Q) \rightarrow (Q \rightarrow R))$
(A) S is neither a tautology nor a contradiction
(B) S is a tautology
(C) S is a contradiction
(D) The antecedent of S is logically equivalent to the consequent of S.

Solution: (B), (D)

Applications of satisfiability:

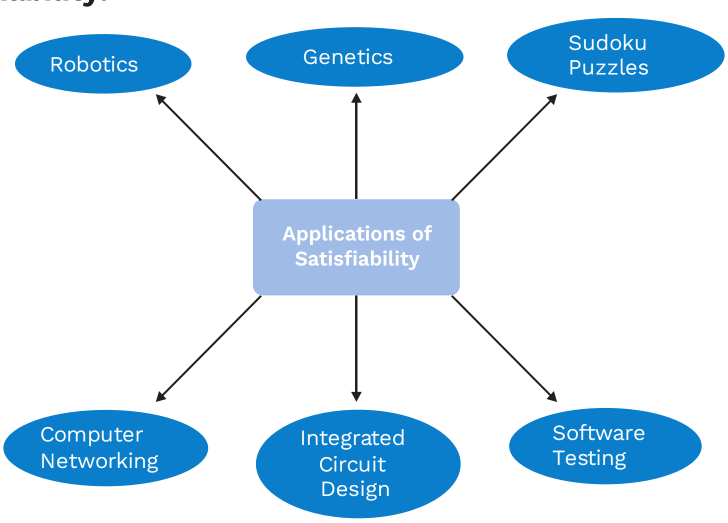


Fig. 1.10

Rack Your Brain

Which of the compound propositions are satisfiable.

(A) $(p \vee q \vee \neg r) \wedge (p \vee \neg q \vee \neg s) \wedge (p \vee \neg r \vee \neg s) \wedge (\neg p \vee \neg q \vee \neg s) \wedge (p \vee q \vee \neg s)$

(B) $(\neg p \vee \neg q \vee r) \wedge (\neg p \vee q \vee \neg s) \wedge (p \vee \neg q \vee \neg s) \wedge (\neg p \vee \neg r \vee \neg s) \wedge (p \vee q \vee \neg r) \wedge (p \vee \neg r \vee \neg s)$

Logical equivalence:

Equivalences (laws):

T = Compound statement that is always true.

F = Compound statement that is always false.

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

Table 11 The Special Case of Boolean Algebra Identities

These laws help to make complicated-looking boolean expressions simple.

"Bitty bought a butter, but the butter was bitter, so bitty bought another butter to make bitter butter, better butter."

- As, the above statement seems a little complicated, but it can be made simpler using English.
- Similarly, in mathematical logic, complicated boolean expressions can be made simpler with the laws stated above in table 11.

Equivalences involving implication:

$f \to s$	$\equiv$	$\neg f \vee s$
$f \to s$	$\equiv$	$\neg s \to \neg f$
$f \vee s$	$\equiv$	$\neg f \to s$
$\neg (f \to s)$	$\equiv$	$f \wedge \neg s$
$(f \to s) \wedge (f \to r)$	$\equiv$	$f \to (s \wedge r)$
$(f \to r) \wedge (s \to r)$	$\equiv$	$(f \vee s) \to r$
$(f \to s) \vee (f \to r)$	$\equiv$	$f \to (s \vee r)$
$(f \to r) \vee (s \to r)$	$\equiv$	$(f \vee s) \to r$

Table 12 Logical Equivalences Involving Conditional Statements

$f \leftrightarrow s$	$\equiv$	$(f \to s) \wedge (s \to f)$
$f \leftrightarrow s$	$\equiv$	$\neg f \leftrightarrow \neg s$
$f \leftrightarrow s$	$\equiv$	$(f \wedge s) \vee (\neg f \vee \neg s)$
$\neg (f \leftrightarrow s)$	$\equiv$	$f \leftrightarrow \neg s$

Table 13 Logical Equivalence Involving Biconditional Statements

De Morgan's laws:

As we have already seen that De Morgan's laws state:

- $\neg (p \wedge q) \equiv \neg p \vee \neg q$
- $\neg (p \wedge q) \equiv \neg p \wedge \neg q$

Note:

When using De Morgan's laws, remember to change the logical connective after you negate.

It states that negation of conjunction/disjunction is formed by disjunction/

conjunction of negation of the component propositions.

4. Use De Morgan's law to express the negation of "Maya will go to fare or Abdul will go to fare".

Solution:

⇒ Let p = Maya will go to fare

q = Abdul will go to fare

Can be represented by $p \vee q$

Now, the negation of $p \vee q = \neg (p \vee q)$

According to De Morgan's law $\neg (p \vee q) = \neg p \wedge \neg q$

Which states that Maya will not go to fare and Abdul will not go to the fare.

5. Use De-Morgan's law, to show $\neg (p \vee (\neg p \wedge q))$ and $\neg p \wedge \neg q$ are logical equivalent.

Solution:

$\neg (p \vee (\neg p \wedge q))$ [using De Morgan law]

$\neg p \wedge \neg (\neg p \wedge q)$

$\neg p \wedge [\neg (\neg p) \vee \neg q]$

$\neg P \wedge (p \vee \neg q)$ [Using distributive law]

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

$F \vee (\neg p \wedge \neg q)$ $(\because \neg p \wedge p \equiv F)$

$\neg p \wedge \neg q$ [using commutative law of disjunction]

Hence, proved, $\neg (p \vee (\neg p \wedge q))$ is logically equivalent to $\neg p \wedge \neg q$.

The general form of an argument: (Inference)

The process of deriving the conclusion based on assumption is called an **argument**.

The conjunction of premises implies a conclusion.

$(p_1 \wedge p_2 \wedge ...p_n) \rightarrow q$

An inference which is **tautology** called **valid inference** otherwise invalid inference.

Rule of inference:

Any valid inference is the rule of inference.

Name	Rule of Inference	Tautology
Addition	$\dfrac{p}{\therefore p \vee q}$	$p \rightarrow (p \vee q)$
Conjunction	$\dfrac{q}{\therefore p \wedge q}$	$([p] \wedge [q]) \rightarrow (p \wedge q)$
Simplification	$\dfrac{p \wedge q}{\therefore p}$	$(p \wedge q) \rightarrow p$
Modus ponens	$p \rightarrow q$ $\dfrac{p}{\therefore q}$	$(p \wedge [P \rightarrow q]) \rightarrow q$
Hypothetical syllogism	$p \rightarrow q$ $\dfrac{q \rightarrow r}{\therefore p \rightarrow r}$	$([p \rightarrow q) \wedge (q \rightarrow r]) \rightarrow (p \rightarrow r)$
Disjunctive syllogism	$p \vee q$ $\dfrac{\neg p}{\therefore q}$	$([p \vee q) \neg p] \rightarrow q$
Modus tollens	$p \rightarrow q$ $\dfrac{\neg q}{\therefore \neg p}$	$(\neg q \wedge [p \rightarrow q]) \rightarrow \neg p$
Resolution	$p \vee q$ $\dfrac{\neg p \vee r}{\therefore q \vee r}$	$([p \vee q) \wedge (\neg p \vee r]) \rightarrow (q \vee r)$

Table 14 Rules of Inference

6. "If Vinay comes to the ceremony, Atul will not come to the ceremony. If Atul doesn't come to the ceremony, Siddhu will come to the ceremony."

Solution:

Let the propositions be as follows:

p: Vinay comes to the ceremony.

q: Atul does not come to the ceremony.

r: Siddhu comes to the ceremony.

$$\begin{array}{l} \therefore p \rightarrow q \\ \quad q \rightarrow r \\ \hline \therefore p \rightarrow r \end{array}$$

This argument is a hypothetical syllogism.

Functional completeness: If any boolean function can be expressed using a given set of boolean functions, then that set of boolean functions is functionally complete.

For example, The set {∧, ∨, ¬} is clearly functionally complete.

> **Note:**
>
> With n variable, $2^{(2^n)}$ boolean functions can be represented.

- The set {∧, ¬} is said to be functionally complete or minimal functionally complete set.
- The set {∨, ¬} is also functionally complete.
- The set {∧, ∨} is not functionally complete as we can not generate "not" with the of "AND" and "OR".

> **Note:**
>
> A set is said to be functionally complete if we can derive a set which is already functionally complete.

Minimally functionally complete set: A set is said to be minimally functionally complete if:

- It is functionally complete
- No subset of the given set is functionally complete.

For example:

- {∧, ∨, ¬} is not minimally functionally complete.
- {∧, ¬}, {∨, ¬} are minimally functionally complete.

  E.g., with ↑ we can generate "NOT", "AND", and "OR".

> **Note:**
>
> {↑} and {↓} are smallest functionally complete sets.

Normal forms: The method of reducing a given formula to an equivalent form is called 'normal form'.

There are two types of standard normal forms:

1. PDNF (Principal disjunctive normal form)
2. PCNF (Principal conjuctive normal form)

Considering m for minterm and M for maxterm.

> **Note:**
>
> Number of terms in PDNF + Number of terms in PCNF = 2^n

After solving the boolean function.

If the conclusion is '1' then it'll be considered in minterms maxterm otherwise.

PDNF: Disjunction of min terms

PCNF: Conjunction of max terms

E.g., $p \leftrightarrow (q \rightarrow r')$ $\begin{bmatrix} 0 = \text{False} \\ 1 = \text{True} \end{bmatrix}$

p	q	r	q → r'	p ↔ (q → r')	Min Terms	Max Terms
0	0	0	1	0	m_0	$M_0 = p \vee q \vee r$
0	0	1	1	0	m_1	$M_1 = p \vee q \vee r'$
0	1	0	1	0	m_2	$M_2 = p \vee q' \vee r$

0	1	1	0	1	$m_3 = p' \wedge q \wedge r$	M_3
1	0	0	1	1	$m_4 = p \wedge q' \wedge r'$	M_4
1	0	1	1	1	$m_5 = p \wedge q' \wedge r'$	M_5
1	1	0	1	1	$m_6 = p \wedge q \wedge r'$	M_6
1	1	1	0	0	m_7	$M_7 = p' \vee q' \vee r'$

Table 15

Predicates and Quantifies

The meaning of the English statement may not always be possible to express in the form of propositional logic.

For e.g., "Every computer science student in the university is intelligent."

Now, with the help of propositional logic, we in no way can prove Sanya is intelligent.

Where Sanya is one of the computer science students at the university.

Place where propositional logic can not work, predicate logic comes into the picture. To understand predicate logic, let's first learn about predicate properly.

Predicates:

Consider statement, " x is greater than 5".

- The subject part: Variable itself
- The predicate part: Is greater than 5

so we can denote this statement as P(x), where P is the predicate part and x is variable. When the value is assigned to the variable predicate is converted to propositional logic.

Rack Your Brain

Let p(x) denotes the statement "$x \leq 4$". What are these truth values?

(A) p(0)

(B) p(4)

(C) p(6)

Pre conditions and post conditions:

Statements that describe valid input are known as **pre-conditions**, and the condition that the output should satisfy when the program has run is called **post-conditions**.

Quantifiers:

Quantification is a way to create a proposition from a propositional function.

Note:
The area of logic that deals with predicate and quantifiers is called predicate logic.

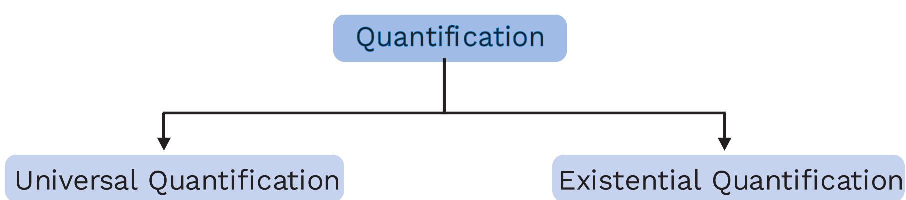


Fig. 1.11

Universal quantifier:

- It tells us the predicate is true for every element under consideration.
- To understand universal quantifier; first we need to understand the term domain of discourse/universe of discourse/domain:
  When all the values of a variable in a particular domain are true, then a property is called true, that particular domain is called the domain of discourse/universe of discourse/domain.
  These types of statements are expressed using universal quantification.

Definition

These universal Quantification of P(x) is the statement.
"P(x) for all values of x in the domain"
The notation $\forall xP(x)$, denotes the universal Quantification of P(x). Here $\forall x$ is called the universal Quantifier. We read $\forall xP(x)$ as "for all xP(x)" or "for every xP(x)". An element for which P(x) is false is called a counter example of $\forall xP(x)$.

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

Table 16

Existential quantifier:

- By existential quantifier we know that there is atleast one value for which predicate is true.
- In existential quantification, a proposition is formed that is true if and only if P(x) is true for at least one value of x in the domain.

Note:

Truth value of xP(x) depends on the domain.

The uniqueness quantifier:

There is one more type of quantifier called "uniqueness quantifier" denoted by $\exists!$ or $\exists_1$.

The notation $\exists! xP(x)$ or $\exists_1 xP(x)$ states, "There exists a unique x such that P(x) is true".

Note:
The truth value of xP(x) depends on the domain.

Logical equivalences involving quantifiers:
Standard definition:

Definition

If two statements have same truth table, then both statements are logically equivalent that two statements S and T involving predicates and quantifiers are logically equivalent.

7. Consider the following statements:
   I. p q p q is logically equivalent.
   II. p q (p q) (p q) is logically equivalent.
   Which of the following options are correct?
   (A) Only I is true.
   (B) Only II is true.
   (C) Both are true.
   (D) None of the above.

Solution: (C)

I.

p	**q**	**$p \to q$**
0	0	1
0	1	1
1	0	0
1	1	1

Table 1.17

p	**q**	**$\neg p$**	**$\neg p \vee q$**
0	0	1	1
0	1	1	1
1	0	0	0
1	1	0	1

Table 1.18

Now, $p \rightarrow q$ and $\neg p \vee q$ are precisely the same.

$\therefore$ Both boolean expressions as logically equivalent.

II.

p	q	$p \leftrightarrow q$
0	0	1
0	1	0
1	0	0
1	1	1

Table 1.19

p	q	$\neg p$	$\neg q$	$(p \wedge q)\ (\neg p \neg q)$
0	0	1	1	1
0	1	1	0	0
1	0	0	1	0
1	1	0	0	1

Table 1.20

We can clearly see the column $p \leftrightarrow q$ and $(p \wedge q) \vee (\neg p \wedge \neg q)$ have same values, which means these two expressions are logically equivalent.

Note:

- The quantifiers $\forall$ and $\exists$ have a higher precedence than all logical operators.
- When a quantifier is used on the variable x, this occurrence of a variable is Bound.
- The part of the logical expression to which quantifier is applied is called scope.

De Morgan's Laws for Quantifiers			
Negation	**Equivalent Statement**	**Why True**	**Why False**
$\neg\exists x P(x)$	$\forall x \neg P(x)$	For every x, P(x) is false.	There is an x for which P(x) is true.
$\neg\forall x P(x)$	$\exists x \neg P(x)$	There is an x for which P(x) is false.	P(x) is true for every x.

Table 1.21 De Morgan's laws for Quantifiers

Aristotle form:

1. All p's are Q's $\quad \forall x[p(x) \rightarrow Q(x)]$
2. Some p's are Q's $\quad \exists x[p(x) \wedge Q(x)]$
3. Not all p's are Q's $\quad \sim \forall x[p(x) \rightarrow Q(x)]$
   $\equiv \exists x[p(x) \wedge \sim Q(x)]$
   $\equiv$ Some p's are not Q's
4. No p's are Q's $\quad \sim\exists x[p(x) \wedge Q(x)]$
   $\equiv \forall x[p(x) \rightarrow \sim Q(x)]$
   All p's are not Q's

Note:

$\forall$ follow implication $(\rightarrow)$

$\exists$ follow and $(\vee)$.

Rack Your Brain

Some real no's are not rational.

1. $\exists x[\text{real}(x) \vee \text{rational}(x)]$
2. $\exists x(\text{real}(x) \rightarrow \text{rational}(x)]$
3. $\sim\forall x[\text{real}(x) \wedge \sim\text{rational}(x)]$
4. $\exists x[\text{rational}(x) \rightarrow \text{real}(x)]$

8. Match List I and List II

List I

(A) Everyone loves Obama

(B) Everyone loves someone

(C) There is someone whom everyone loves

(D) There is someone whom no one loves

List II

1. loves (x, Obama)
2. y loves (x,y)
3. y loves (x,y
4. y loves (x,y)

	A	B	C	D
1.	1	2	3	4
2.	1	3	2	4
3.	1	4	3	2
4.	1	2	4	3

Solution:

Everyone loves Obama	:	$\forall x$loves (x, Obama)
Everyone loves someone	:	$\forall x\ \exists y$ loves (x,y)
There is someone whom everyone	:	$\exists y\ \forall x$ loves (x,y)
There is someone whom no one loves	:	$\exists y\ \forall x\ \neg$loves (x,y)

Previous Years' Questions

Choose the correct translation of given statement: **[GATE IT 2013]**

"None of my friends are perfect".

(A) x(F(x) p(x)) (B) x(F(x) p(x))

(C) x(F(x) p(x)) (D) x(F(x) p(x))

Solution: (D)

Nested quantifiers:

We use quantifiers to express mathematical statements such as "The sum of two positive integers is always positive".

Note:

Be careful with the order of existential and universal quantifier.

Statement	When True	When False
$\forall x \forall y$ P(x, y) $\forall y \forall x$ P(x, y)	P(x, y) is true for every pair x, y.	There is a pair x, y for which P(x, y) is false.
$\forall x \exists y$ P(x, y)	For every x, there is a y for which P(x, y) is true.	There is an x such that P(x, y) is false for every y.
$\exists x\ \forall y$ P(x, y)	There is an x for which P(x, y) is true for every y.	For every x, there is a y for which P(x, y) is false.
$\exists x \exists y$ P(x, y) $\exists y \exists x$ P(x, y)	There is a pair x, y for which P(x, y) is true.	P(x, y) is false for every pair x, y.

Table 1.22 Quantification of Two Variables

9. Translate the statement $\forall x(C(x) \vee \exists\ y(C(y) \wedge F(x,y)))$ In English, where P(x) is "x has a pentab", C(x,y) is "x and y are colleagues" and the domain for both x and y consists of all the teachers in a college.

Solution:

The statement says that for every teacher x in the college, x has a pentab or there is a teacher y such that y has a pentab and x and y are colleagues. In other words, every teacher in the college has a pentab or has a colleague who has a pentab.

10. Which of the following options is the correct expression in predicates and quantifiers "A student must take at least 60 hours course or at least 45 hours course and write a master's thesis, and receive a grade not lower than

'B' in all required courses, to receive master's degree".

(A) M → ((H(60) ∧ (H(45) ∧ T)) ∨∀yG(B,y)

(B) M → ((H(60) ∨ (H(45) ∧ T)) ∨∀yG(B,y)

(C) M → ((H(60) ∧ (H(45) ∨ T)) ∨∀yG(B,y)

(D) M → ((H(60) ∧ (H(45) ∨ T)) ∨∀yG(B,y)

Solution: (B)

M→((H(60)∨(H(45)∧T))∨∀yG(B, y)

Where M is the proposition "The student receiver master's degree".

H(x) is "The student took at least x hours course".

T is the proposition "The student wrote a thesis".

G(x, y) is "The person got grade x or higher in Course G".

Previous Years' Questions

Which of the following is negation of:

[GATE CSE 2008]

[∀ x, α → (∃y, β → (∀u, ∃v, y))]

(A) [∃x, α → (∀y, β → (∃u, ∀v, y))]

(B) [∃ x, α → (∀y, β → (∃u, ∀v, ¬y))]

(C) [∀x, ¬ α → (∃y, ¬ β → (∀u, ∃v, ¬y))]

(D) [∃x, α∧ (∀y, β ∧(∃u, ∀v, ¬y))]

Solution: (D)

Previous Years' Questions

Consider the following formula and its two interpretations I_1 and I_2.

[GATE CSE 2003]

α: (∀x) [P_x ↔ (∀y) [Q_{xy} ↔ Q_{yy}]] → (∀x) [¬ P_x]

l1: Domain: The set of natural numbers.

P_x ≡ xis a prime number

Q_{xy} ≡ ydivides x

l_2: same as l_1 except that p_x= x is a composite number.

Which of the following is true?

(A) l satisfies , l does not

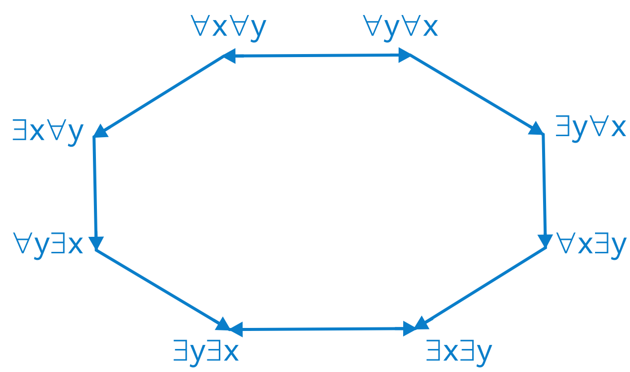
(B) l satisfies , l does not

(C) Neither l nor l satisfies

(D) Both l and l satisfies

Solution: (D)

Relation between two-place predicate:



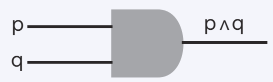
From the above diagram, we can define the following predicate property and many more according to direction.

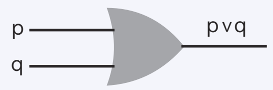
1. ∀x ∀y p(x,y) ≡ ∀y ∀x p(x,y)
2. ∀x ∀y p(x,y) → ∃x ∀y p(x,y)
3. ∃x ∀y p(x,y) → ∀y ∃x p(x,y)
4. ∀x ∀y p(x,y) → ∃y ∀x p(x,y)
5. ∃y ∀x p(x,y) → ∀x ∃y p(x,y)
6. ∀x ∃y p(x,y) → ∃y ∃x p(x,y)
7. ∀y ∀x xp(x,y) → ∃x ∀y p(x,y)
8. ∀y ∃x p(x,y) → ∃y ∃x p(x,y)
9. ∀y ∃x p(x,y) → ∃x ∃y p(x,y)
10. ∃y ∃x p(x,y) ≡ ∃x ∃y p(x,y)

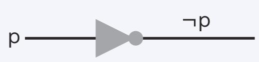
Quantifiers property:

1. ∀x [p(x) ∧ Q(x)] ≡ ∀x p(x) ∧ ∀x Q(x)
2. ∃x[p(x) ∨ Q(x)] ≡ ∃x p(x) ∨ ∃x Q(x)
3. ∀x p(x) ∨ ∀x Q(x) → ∀x[p(x) ∨Q(x)]
4. ∃x[p(x) ∧ Q(x)] → ∃x p(x) ∧ ∃x Q(x)
5. ∀x[p(x) ∧ Q] ≡ ∀x p(x) ∧ Q
6. ∀x[p(x) ∨ Q] ≡ ∀x p(x) ∨ Q
7. ∃x[p(x) ∨ Q] ≡ ∃x p(x) ∨ Q
8. ∃x[p(x) ∧ Q] ≡ ∃x p(x) ∧ Q
9. ∀x[p→Q(x)] ≡ p→∀x Q(x)
10. ∃x[p → Q(x)] ≡ p→ ∃x Q(x)
11. ∀x[p(x) → Q] ≡ ∃ p(x) → Q
12. ∃x[p(x) → Q] ≡ ∀x p(x) → Q

Chapter Summary

- Propositional logic is a declarative statement which results in either true or false.
- There are two types of propositions:
  - Atomic proposition
  - Compound proposition
- Connectives: The weak point at which a compound statement is breakable is called connective.
- There are four types of connectives:
  - Conjunction ($\wedge$)
  - Disjunction ($\vee$/$\oplus$)
  - Simple implication ($\rightarrow$)
  - Double implication ($\leftrightarrow$)
- AND gate: 

- OR gate: 

- NOT gate: 

- $\forall$ is called as universal quantifier.
- $\exists$ is called as existential quantifier.
- There are two types of argument:
  - Valid
  - Invalid
- A proposition is valid if each disjunctive clause in any CNF representation of proposition contains a pair of complementary literals.
- PCNF and PDNF are unique.
- If a = b, then PDNF of a and b will be same, and PCNF will also be same.