**LECTURE NOTES**

**UNIT -I:**

**Mathematical Logic**:

Propositional Calculus: Statements and Notations, Connectives, Well Formed Formulas, Truth Tables, Tautologies, Equivalence of Formulas, Duality Law, Tautological Implications,

**INTRODUCTION**

**Unit – I**

**Mathematical Logic**

**Proposition:** A **proposition** or **statement** is a declarative sentence which is either true or false but not both. The truth or falsity of a proposition is called its **truth-value.**

These two values ‗true‘ and ‗false‘ are denoted by the symbols *T* and *F*

respectively. Sometimes these are also denoted by the symbols 1 and 0 respectively.

**Example 1:** Consider the following sentences:

* 1. Delhi is the capital of India.
  2. Kolkata is a country.

3. 5 is a prime number. 4. 2 + 3 = 4.

These are propositions (or statements) because they are either true of false. Next consider the following sentences:

1. How beautiful are you?
2. Wish you a happy new year
3. *x* + *y* = *z*
4. Take one book.

These are not propositions as they are not declarative in nature, that is, they do not declare a definite truth value *T* or *F*.

**Propositional Calculus** is also known as **statement calculus.** It is the branch of mathematics that is used to describe a logical system or structure. A logical system consists of (1) a universe of propositions, (2) truth tables (as axioms) for the logical operators and (3) definitions that explain equivalence and implication of propositions.

**Connectives**

The words or phrases or symbols which are used to make a proposition by two or more propositions are called **logical connectives** or **simply connectives.** There are five basic connectives called negation, conjunction, disjunction, conditional and biconditional.

**Negation**

The **negation** of a statement is generally formed by writing the word ‗not‘ at a proper place in the statement (proposition) or by prefixing the statement with the phrase

‗It is not the case that‘. If *p* denotes a statement then the negation of *p* is written as *p* and read as ‗not *p*‘. If the truth value of *p* is *T* then the truth value of *p* is *F*. Also if the truth value of *p* is *F* then the truth value of *p* is *T*.

**Table 1.** Truth table for negation

|  |  |
| --- | --- |
| p | ¬p |
| T F | F T |

**Example 2:** Consider the statement *p*: Kolkata is a city. Then ¬p: Kolkata is not a city.

Although the two statements ‗Kolkata is not a city‘ and ‗It is not the case that Kolkata is a city‘ are not identical, we have translated both of them by *p*. The reason is that both these statements have the same meaning.

**Conjunction**

The **conjunction** of two statements (or propositions) *p* and *q* is the statement *p* ∧ *q* which is read as ‗*p* and *q*‘. The statement *p* ∧ *q* has the truth value *T* whenever both *p* and *q* have the truth value *T*. Otherwise it has truth value *F*.

**Table 2.** Truth table for conjunction

|  |  |  |  |
| --- | --- | --- | --- |
| *p* | *q* | *p* ∧ | *q* |
| *T* | *T* | *T* | |
| *T* | *F* | *F* | |
| *F* | *T* | *F* | |
| *F* | *F* | *F* | |

**Example 3:** Consider the following statements *p* : It is raining today.

*q* : There are 10 chairs in the room.

Then *p* ∧ *q* : It is raining today and there are 10 chairs in the room.

**Note:** Usually, in our everyday language the conjunction ‗and‘ is used between two statements which have some kind of relation. Thus a statement ‗It is raining today and 1 + 1 = 2‘ sounds odd, but in logic it is a perfectly acceptable statement formed from the statements ‗It is raining today‘ and ‗1 + 1 = 2‘.

**Example 4:** Translate the following statement:

‗Jack and Jill went up the hill‘ into symbolic form using conjunction.

**Solution:** Let p : Jack went up the hill, q : Jill went up the hill.

Then the given statement can be written in symbolic form as p ∧ q.

**Disjunction**

The **disjunction** of two statements p and q is the statement p ∨ q which is read as ‗p or q‘. The statement p ∨ q has the truth value F only when both p and q have the truth value F. Otherwise it has truth value T.

**Table 3:** Truth table for disjunction

|  |  |  |  |
| --- | --- | --- | --- |
| *p* | *q* | *p* ∨ | *q* |
| *T* | *T* | *T* | |
| *T* | *F* | *T* | |
| *F* | *T* | *T* | |
| *F* | *F* | *F* | |

**Example 5:** Consider the following statements *p* : I shall go to the game.

*q* : I shall watch the game on television.

Then *p* ∨ *q* : I shall go to the game or watch the game on television.

**Conditional proposition**

If *p* and *q* are any two statements (or propositions) then the statement *p* → *q* which is read as,

‗If *p*, then *q*‘ is called a **conditional statement** (or **proposition**) or **implication** and the connective is the **conditional connective.**

The conditional is defined by the following table:

**Table 4.** Truth table for conditional

|  |  |  |  |
| --- | --- | --- | --- |
| *p* | *q* | *p* → | *q* |
| *T* | *T* | *T* | |
| *T* | *F* | *F* | |
| *F* | *T* | *T* | |
| *F* | *F* | *T* | |

In this conditional statement, *p* is called the **hypothesis** or **premise** or **antecedent** and *q* is called the **consequence** or **conclusion.**

To understand better, this connective can be looked as a conditional promise. If the promise is violated (broken), the conditional (implication) is false. Otherwise it is true. For this reason, the only circumstances under which the conditional *p* → *q* is false is when *p* is true and *q* is false.

**Example 6:** *Translate the following statement:*

*‘The crop will be destroyed if there is a flood’ into symbolic form using conditional connective.*

**Solution:** Let *c* : the crop will be destroyed; *f* : there is a flood. Let us rewrite the given statement as

‗If there is a flood, then the crop will be destroyed‘. So, the symbolic form of the given statement is *f* → *c*.

**Example 7:** Let p and q denote the statements: p : You drive over 70 km per hour.

q : You get a speeding ticket.

Write the following statements into symbolic forms.

1. You will get a speeding ticket if you drive over 70 km per hour.
2. Driving over 70 km per hour is sufficient for getting a speeding ticket.
3. If you do not drive over 70 km per hour then you will not get a speeding ticket.
4. Whenever you get a speeding ticket, you drive over 70 km per hour.

**Solution:** (i) p → q (ii) p → q (iii) p → q (iv) q → p.

**Notes:** 1. In ordinary language, it is customary to assume some kind of relationship between the antecedent and the consequent in using the conditional. But in logic, the antecedent and the

consequent in a conditional statement are not required to refer to the same subject matter. For example, the statement ‗If I get sufficient money then I shall purchase a high-speed computer‘ sounds reasonable. On the other hand, a statement such as ‗If I purchase a computer then this pen is red‘ does not make sense in our conventional language. But according to the definition of conditional, this proposition is perfectly acceptable and has a truth-value which depends on the truth-values of the component statements.

1. Some of the alternative terminologies used to express *p* → *q* (if *p*, then *q*) are the following: (*i*) *p* implies *q*
2. *p* only if *q* (‗If *p*, then *q*‘ formulation emphasizes the antecedent, whereas ‗*p* only if *q*‘ formulation emphasizes the consequent. The difference is only stylistic.)
3. *q* if *p*, or *q* when *p*.
4. *q* follows from *p*, or *q* whenever *p*.
5. *p* is sufficient for *q*, or a sufficient condition for *q* is *p*. (*vi*) *q* is necessary for *p*, or a necessary condition for *p* is *q*. (*vii*) *q* is consequence of *p*.

**Converse, Inverse and Contrapositive**

If *P → Q* is a conditional statement, then (1). *Q → P* is called its *converse*

* 1. *¬P → ¬Q* is called its *inverse*
  2. *¬Q → ¬P* is called its *contrapositive*. Truth table for *Q → P* (converse of *P → Q*)

|  |  |  |
| --- | --- | --- |
| *P* | *Q* | *Q → P* |
| T | T | T |
| T | F | T |
| F | T | F |
| F | F | T |

Truth table for *¬P → ¬Q* (inverse of *P → Q*)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *P* | *Q* | *¬P* | *¬Q* | *¬P → ¬Q* |
| T | T | F | F | T |
| T | F | F | T | T |
| F | T | T | F | F |
| F | F | T | T | T |

Truth table for *¬Q → ¬P* (contrapositive of *P → Q*)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *P* | *Q* | *¬Q* | *¬P* | *¬Q →* | *¬P* |
| T | T | F | F | T | |
| T | F | T | F | F | |
| F | T | F | T | T | |
| F | F | T | T | T | |

**Example:** Consider the statement

*P* : It rains.

*Q*: The crop will grow. The implication *P → Q* states that

*R*: If it rains then the crop will grow.

The converse of the implication *P → Q*, namely *Q → P* sates that *S*: If the crop will grow then there has been rain.

The inverse of the implication *P → Q*, namely *¬P → ¬Q* sates that

*U*: If it does not rain then the crop will not grow.

The contraposition of the implication *P → Q*, namely *¬Q → ¬P* states that *T* : If the crop do not grow then there has been no rain.

**Example 9:** *Construct the truth table for (p* → *q)* ∧ *(q* →p)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *p* | *q* | *p* → *q* | *q* → *p* | *(p* → *q)* ∧ *(q* → *p)* |
|  |  |  |  |  |
| *T* | *T* | *T* | *T* | *T* |
| *T* | *F* | *F* | *T* | *F* |
| *F* | *T* | *T* | *F* | *F* |
| *F* | *F* | *T* | *T* | *T* |

**Biconditional proposition**

If *p* and *q* are any two statements (propositions), then the statement *p↔ q* which is read as ‗*p* if and only if *q*‘ and abbreviated as ‗*p* iff *q*‘ is called a **biconditional statement** and the connective is the **biconditional connective.**

The truth table of p*↔*q is given by the following table:

**Table 6.** Truth table for biconditional

|  |  |  |
| --- | --- | --- |
| *p* | *q* | p*↔*q |
| *T* | *T* | *T* |
| *T* | *F* | *F* |
| *F* | *T* | *F* |
| *F* | *F* | *T* |

It may be noted that *p q* is true only when both *p* and *q* are true or when both *p* and *q* are false. Observe that *p q* is true when both the conditionals *p* → *q* and *q* → *p* are true, *i.e.*, the truth- values of (*p* → *q*) ∧ (*q* → *p*), given in Ex. 9, are identical to the truth-values of *p q* defined here.

**Note:** The notation *p* ↔ *q* is also used instead of p*↔*q.

**TAUTOLOGY AND CONTRADICTION**

**Tautology:** A statement formula which is true regardless of the truth values of the statements which replace the variables in it is called a **universally valid formula** or a **logical truth** or a **tautology.**

**Contradiction:** A statement formula which is false regardless of the truth values of the statements which replace the variables in it is said to be a **contradiction.**

**Contingency:** A statement formula which is neither a tautology nor a contradiction is known as a **contingency.**

**Substitution Instance**

A formula *A* is called a substitution instance of another formula *B* if *A* can be obtained form *B* by substituting formulas for some variables of *B*, with the condition that the same formula is substituted for the same variable each time it occurs.

Example: Let *B* : *P →* (*J* ∧ *P* )*.*

Substitute *R↔S* for *P* in *B*, we get

*(i)*: (*R ↔ S*) *→* (*J* ∧ (*R ↔ S*))

Then *A* is a substitution instance of *B*.

Note that (*R ↔ S*) *→* (*J* ∧*P*) is not a substitution instance of *B* because the variables P in *J* ∧ *P* was not replaced by *R ↔ S*.

**Equivalence of Formulas**

Two formulas *A* and *B* are said to equivalent to each other if and only if *A↔ B* is a tautology.

If *A↔B* is a tautology, we write *A* ⇔ *B* which is read as *A* is equivalent to *B*.

Note : 1. ⇔ is only symbol, but not connective.

* 1. *A ↔ B* is a tautology if and only if truth tables of *A* and *B* are the same.
  2. Equivalence relation is symmetric and transitive.

Method I. Truth Table Method: One method to determine whether any two statement formulas are equivalent is to construct their truth tables.

Example: Prove *P* ∨ *Q* ⇔ *¬*(*¬P* ∧ *¬Q*). Solution:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *P* | *Q* | *P* ∨ *Q* | *¬P* | *¬Q* | *¬P* ∧ *¬Q* | *¬*(*¬P* ∧ *¬Q*) | (*P* ∨ *Q*) ⇔ *¬*(*¬P* ∧ *¬Q*) |
| T | T | T | F | F | F | T | T |
| T | F | T | F | T | F | T | T |
| F | T | T | T | F | F | T | T |
| F | F | F | T | T | T | F | T |

As *P* ∨ *Q ¬*(*¬P* ∧ *¬Q*) is a tautology, then *P* ∨ *Q* ⇔ *¬*(*¬P* ∧ *¬Q*). Example: Prove (*P → Q*) ⇔ (*¬P* ∨ *Q*).

Solution:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *P* | *Q* | *P → Q* | *¬P* | *¬P* ∨ *Q* | (*P → Q*) (*¬P* ∨ *Q*) |
| T | T | T | F | T | T |
| T | F | F | F | F | T |
| F | T | T | T | T | T |
| F | F | T | T | T | T |

As (*P → Q*) (*¬P* ∨ *Q*) is a tautology then (*P → Q*) ⇔ (*¬P* ∨ *Q*).

Equivalence Formulas:

1. Idempotent laws:
   1. *P* ∨ *P* ⇔ *P* (b) *P* ∧ *P* ⇔ *P*
2. Associative laws:

(a) (*P* ∨ *Q*) ∨ *R* ⇔ *P* ∨ (*Q* ∨ *R*) (b) (*P* ∧ *Q*) ∧ *R* ⇔ *P* ∧ (*Q* ∧ *R*)

1. Commutative laws:
   1. *P* ∨ *Q* ⇔ *Q* ∨ *P* (b) *P* ∧ *Q* ⇔ *Q* ∧ *P*
2. Distributive laws:

*P* ∨ (*Q* ∧ *R*) ⇔ (*P* ∨ *Q*) ∧ (*P* ∨ *R*) *P* ∧ (*Q* ∨ *R*) ⇔ (*P* ∧ *Q*) ∨ (*P* ∧ *R*)

1. Identity laws:
   1. (i) *P* ∨ *F* ⇔ *P* (ii) *P* ∨ *T* ⇔ *T*
   2. (i) *P* ∧ *T* ⇔ *P* (ii) *P* ∧ *F* ⇔ *F*
2. Component laws:

|  |  |  |
| --- | --- | --- |
| (a) (i) *P* ∨ *¬P* ⇔ *T* | (ii) *P* ∧ *¬P* ⇔ *F* | . |
| (b) (i) *¬¬P* ⇔ *P*  7. Absorption laws: | (ii) *¬T* ⇔ *F* , *¬F* ⇔ *T* |  |

(a) *P* ∨ (*P* ∧ *Q*) ⇔ *P* (b) *P* ∧ (*P* ∨ *Q*) ⇔ *P*

8. Demorgan‘s laws:

(a) *¬*(*P* ∨ *Q*) ⇔ *¬P* ∧ *¬Q* (b) *¬*(*P* ∧ *Q*) ⇔ *¬P* ∨ *¬Q*

**Method II. Replacement Process**: Consider a formula *A* : *P →* (*Q → R*). The formula *Q → R* is a part of the formula *A*. If we replace *Q → R* by an equivalent formula *¬Q*∨*R* in *A*, we get another

formula *B* : *P →* (*¬Q*∨*R*). One can easily verify that the formulas *A* and *B* are equivalent to each other. This process of obtaining *B* from *A* as the replacement process.

Example: Prove that *P →* (*Q → R*) ⇔ *P →* (*¬Q* ∨ *R*) ⇔ (*P* ∧ *Q*) *→ R*.(May. 2010) Solution: *P →* (*Q → R*) ⇔ *P →* (*¬Q* ∨ *R*) [∵ *Q → R* ⇔ *¬Q* ∨ *R*]

⇔ *¬P* ∨ (*¬Q* ∨ *R*) [∵ *P → Q* ⇔ *¬P* ∨ *Q*]

⇔ (*¬P* ∨ *¬Q*) ∨ *R* [by Associative laws]

⇔ *¬*(*P* ∧ *Q*) ∨ *R* [by De Morgan‘s laws]

⇔ (*P* ∧ *Q*) *→ R*[∵ *P → Q* ⇔ *¬P* ∨ *Q*]*.*

Example: Prove that (*P → Q*) ∧ (*R → Q*) ⇔ (*P* ∨ *R*) *→ Q*. Solution: (*P → Q*) ∧ (*R → Q*) ⇔ (*¬P* ∨ *Q*) ∧ (*¬R* ∨ *Q*)

⇔ (*¬P* ∧ *¬R*) ∨ *Q* ⇔

*¬*(*P* ∨ *R*) ∨ *Q* ⇔ *P* ∨

*R → Q*

Example: Prove that *P →* (*Q → P* ) ⇔ *¬P →* (*P → Q*). Solution: *P→* (*Q → P* ) ⇔ *¬P* ∨ (*Q → P* )

⇔ *¬P* ∨ (*¬Q* ∨ *P* )

⇔ (*¬P* ∨ *P* ) ∨ *¬Q*

⇔ *T* ∨ *¬Q*

⇔ *T*

and

*¬P →* (*P → Q*) ⇔ *¬*(*¬P* ) ∨ (*P → Q*)

⇔ *P* ∨ (*¬P* ∨ *Q*) ⇔

(*P* ∨ *¬P* ) ∨ *Q* ⇔ *T*

∨ *Q*

⇔ *T*

So, *P →* (*Q → P* ) ⇔ *¬P →* (*P → Q*).

\*\*\*Example: Prove that (*¬P* ∧ (*¬Q* ∧ *R*)) ∨ (*Q* ∧ *R*) ∨ (*P* ∧ *R*) ⇔ *R.* (Nov. 2009) Solution:

(*¬P* ∧ (*¬Q* ∧ *R*)) ∨ (*Q* ∧ *R*) ∨ (*P* ∧ *R*)

⇔ ((*¬P* ∧ *¬Q*) ∧ *R*) ∨ ((*Q* ∨ *P* ) ∧ *R*) [Associative and Distributive laws]

⇔ (*¬*(*P* ∨ *Q*) ∧ *R*) ∨ ((*Q* ∨ *P* ) ∧ *R*) [De Morgan‘s laws]

⇔ (*¬*(*P* ∨ *Q*) ∨ (*P* ∨ *Q*)) ∧ *R* [Distributive laws]

⇔ *T* ∧ *R* [∵ *¬P* ∨ *P* ⇔ *T* ]

⇔ *R*

\*\*Example: Show ((*P* ∨ *Q*) ∧ *¬*(*¬P* ∧ (*¬Q* ∨ *¬R*))) ∨ (*¬P* ∧ *¬Q*) ∨ (*¬P* ∧ *¬R*) is tautology. Solution: By De Morgan‘s laws, we have

*¬P* ∧ *¬Q* ⇔ *¬*(*P* ∨ *Q*)

*¬P* ∨ *¬R* ⇔ *¬*(*P* ∧ *R*)

Therefore

Also

(*¬P* ∧ *¬Q*) ∨ (*¬P* ∧ *¬R*) ⇔ *¬*(*P* ∨ *Q*) ∨ *¬*(*P* ∧ *R*)

⇔ *¬*((*P* ∨ *Q*) ∧ (*P* ∨ *R*))

*¬*(*¬P* ∧ (*¬Q* ∨ *¬R*)) ⇔ *¬*(*¬P* ∧ *¬*(*Q* ∧ *R*))

⇔ *P* ∨ (*Q* ∧ *R*)

⇔ (*P* ∨ *Q*) ∧ (*P* ∨ *R*)

Hence ((*P* ∨ *Q*) ∧ *¬*(*¬P* ∧ (*¬Q* ∨ *¬R*))) ⇔ (*P* ∨ *Q*) ∧ (*P* ∨ *Q*) ∧ (*P* ∨ *R*)

⇔ (*P* ∨ *Q*) ∧ (*P* ∨ *R*) Thus ((*P* ∨ *Q*) ∧ *¬*(*¬P* ∧ (*¬Q* ∨ *¬R*))) ∨ (*¬P* ∧ *¬Q*) ∨ (*¬P* ∧ *¬R*)

⇔ [(*P* ∨ *Q*) ∧ (*P* ∨ *R*)] ∨ *¬*[(*P* ∨ *Q*) ∧ (*P* ∨ *R*)]

⇔ *T*

Hence the given formula is a tautology.

Example: Show that (*P* ∧ *Q*) *→* (*P* ∨ *Q*) is a tautology. (Nov. 2009) Solution: (*P* ∧ *Q*) *→* (*P* ∨ *Q*) ⇔ *¬*(*P* ∧ *Q*) ∨ (*P* ∨ *Q*) [∵ *P → Q* ⇔ *¬P* ∨ *Q*]

⇔ (*¬P* ∨ *¬Q*) ∨ (*P* ∨ *Q*) [by De Morgan‘s laws]

⇔ (*¬P* ∨ *P* ) ∨ (*¬Q* ∨ *Q*) [by Associative laws and commutative laws]

⇔ (*T* ∨ *T* )[by negation laws]

⇔ *T*

Hence, the result.

Example: Write the negation of the following statements.

(a). Jan will take a job in industry or go to graduate school. (b). James will bicycle or run tomorrow.

1. If the processor is fast then the printer is slow.

Solution: (a). Let *P* : Jan will take a job in industry.

*Q*: Jan will go to graduate school.

The given statement can be written in the symbolic as *P* ∨ *Q*. The negation of *P* ∨ *Q* is given by *¬*(*P* ∨ *Q*).

*¬*(*P* ∨ *Q*) ⇔ *¬P* ∧ *¬Q.*

*¬P* ∧ *¬Q*: Jan will not take a job in industry and he will not go to graduate school. (b). Let *P* : James will bicycle.

*Q*: James will run tomorrow.

The given statement can be written in the symbolic as *P* ∨ *Q*. The negation of *P* ∨ *Q* is given by *¬*(*P* ∨ *Q*).

*¬*(*P* ∨ *Q*) ⇔ *¬P* ∧ *¬Q.*

*¬P* ∧ *¬Q*: James will not bicycle and he will not run tomorrow. (c). Let *P* : The processor is fast.

*Q*: The printer is slow.

The given statement can be written in the symbolic as *P → Q*.

The negation of *P → Q* is given by *¬*(*P → Q*).

*¬*(*P → Q*) ⇔ *¬*(*¬P* ∨ *Q*) ⇔ *P* ∧ *¬Q.*

*P* ∧ *¬Q*: The processor is fast and the printer is fast.

Example: Use Demorgans laws to write the negation of each statement. (a). I want a car and worth a cycle.

(b). My cat stays outside or it makes a mess. (c). I‘ve fallen and I can‘t get up.

(d). You study or you don‘t get a good grade.

Solution: (a). I don‘t want a car or not worth a cycle.

* 1. My cat not stays outside and it does not make a mess.
  2. I have not fallen or I can get up.
  3. You can not study and you get a good grade. Exercises: 1. Write the negation of the following statements. (a). If it is raining, then the game is canceled.

(b). If he studies then he will pass the examination.

Are (*p → q*) *→ r* and *p →* (*q → r*) logically equivalent? Justify your answer by using the rules of logic to simply both expressions and also by using truth tables. Solution: (*p → q*) *→ r* and *p →* (*q → r*) are not logically equivalent because

Method I: Consider

and

(*p → q*) *→ r* ⇔ (*¬p* ∨ *q*) *→ r*

⇔ *¬*(*¬p* ∨ *q*) ∨ *r* ⇔

(*p* ∧ *¬q*) ∨ *r*

⇔ (*p* ∧ *r*) ∨ (*¬q* ∧ *r*)

*p →* (*q → r*) ⇔ *p →* (*¬q* ∨ *r*)

⇔ *¬p* ∨ (*¬q* ∨ *r*) ⇔

*¬p* ∨ *¬q* ∨ *r.*

Method II: (Truth Table Method)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *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 |  | T | T | T |
| T | F | F | F |  | T | T | T |
| F | T | T | T |  | T | T | T |
| F | T | F | T |  | F | F | T |
| F | F | T | T |  | T | T | T |
| F | F | F | T |  | F | T | T |

Here the truth values (columns) of (*p → q*) *→ r* and *p →* (*q → r*) are not identical.

Consider the statement: ‖If you study hard, then you will excel‖. Write its converse, contra positive and logical negation in logic.

**Duality Law**

Two formulas *A* and *A*∗ are said to be *duals* of each other if either one can be obtained from the other by replacing ∧ by ∨ and ∨ by ∧. The connectives ∨ and ∧ are called *duals* of each other. If the

formula *A* contains the special variable *T* or *F* , then *A*∗, its dual is obtained by replacing *T* by *F* and

*F* by *T* in addition to the above mentioned interchanges. Example: Write the dual of the following formulas:

(i). (*P* ∨ *Q*) ∧ *R* (ii). (*P* ∧ *Q*) ∨ *T* (iii). (*P* ∧ *Q*) ∨ (*P* ∨ *¬*(*Q* ∧ *¬S*))

Solution: The duals of the formulas may be written as

(i). (*P* ∧ *Q*) ∨ *R* (ii). (*P* ∨ *Q*) ∧ *F* (iii). (*P* ∨ *Q*) ∧ (*P* ∧ *¬*(*Q* ∨ *¬S*))

Result 1: The negation of the formula is equivalent to its dual in which every variable is replaced by its negation.

We can prove

*¬A*(*P*1*, P*2*, ..., Pn*) ⇔ *A*∗(*¬P*1*, ¬P*2*, ..., ¬Pn*)

Example: Prove that (a). *¬*(*P* ∧ *Q*) *→* (*¬P* ∨ (*¬P* ∨ *Q*)) ⇔ (*¬P* ∨ *Q*) (b). (*P* ∨ *Q*) ∧ (*¬P* ∧ (*¬P* ∧ *Q*)) ⇔ (*¬P* ∧ *Q*)

Solution: (a).*¬*(*P* ∧ *Q*) *→* (*¬P* ∨ (*¬P* ∨ *Q*)) ⇔ (*P* ∧ *Q*) ∨ (*¬P* ∨ (*¬P* ∨ *Q*)) [∵ *P → Q* ⇔ *¬P* ∨ *Q*]

⇔ (*P* ∧ *Q*) ∨ (*¬P* ∨ *Q*)

⇔ (*P* ∧ *Q*) ∨ *¬P* ∨ *Q*

⇔ ((*P*∧ *Q*) ∨ *¬P* )) ∨ *Q*

⇔ ((*P* ∨ *¬P* ) ∧ (*Q* ∨ *¬P* )) ∨ *Q*

⇔ (*T* ∧ (*Q* ∨ *¬P* )) ∨ *Q*

⇔ (*Q* ∨ *¬P* ) ∨ *Q*

⇔ *Q* ∨ *¬P*

⇔ *¬P* ∨ *Q*

(b). From (a) Writing the dual

(*P* ∧ *Q*) ∨ (*¬P* ∨ (*¬P* ∨ *Q*)) ⇔ *¬P* ∨ *Q*

(*P* ∨ *Q*) ∧ (*¬P* ∧ (*¬P* ∧ *Q*)) ⇔ (*¬P* ∧ *Q*)

### Tautological Implications

A statement formula *A* is said to *tautologically imply* a statement *B* if and only if *A → B*

is a tautology.

In this case we write *A* ⇒ *B*, which is read as ‘*A* implies *B*‘.

Note: ⇒ is not a connective, *A* ⇒ *B* is not a statement formula.

*A* ⇒ *B* states that *A → B* is tautology.

Clearly *A* ⇒ *B* guarantees that *B* has a truth value *T* whenever *A* has the truth value *T* .

One can determine whether *A* ⇒ *B* by constructing the truth tables of *A* and *B* in the same manner as was done in the determination of *A* ⇔ *B*. Example: Prove that (*P → Q*) ⇒ (*¬Q → ¬P* ).

Solution:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *P* | *Q* | *¬P* | *¬Q* | *P → Q* | *¬Q → ¬P* | (*P → Q*) *→* (*¬Q → ¬P* ) |
| T | T | F | F | T | T | T |
| T | F | F | T | F | F | T |
| F | T | T | F | T | T | T |
| F | F | T | T | T | T | T |

Since all the entries in the last column are true, (*P → Q*) *→* (*¬Q → ¬P* ) is a tautology.

Hence (*P → Q*) ⇒ (*¬Q → ¬P* ).

In order to show any of the given implications, it is suﬃcient to show that an assignment of the truth value *T* to the antecedent of the corresponding condi-

tional leads to the truth value *T* for the consequent. This procedure guarantees that the conditional becomes tautology, thereby proving the implication.

Example: Prove that *¬Q* ∧ (*P → Q*) ⇒ *¬P* .

Solution: Assume that the antecedent *¬Q* ∧ (*P → Q*) has the truth value *T* , then both *¬Q* and *P → Q* have the truth value *T* , which means that *Q* has the truth value *F* , *P → Q* has the truth value *T* . Hence *P* must have the truth value *F* .

Therefore the consequent *¬P* must have the truth value *T*.

*¬Q* ∧ (*P → Q*) ⇒ *¬P* .

Another method to show *A* ⇒ *B* is to assume that the consequent *B* has the truth value *F* and then show that this assumption leads to *A* having the truth value *F* . Then *A → B* must have the truth value *T* .

Example: Show that *¬*(*P → Q*) ⇒ *P* .

Solution: Assume that *P* has the truth value *F* . When *P* has *F* , *P → Q* has *T* , then *¬*(*P → Q*) has *F*

. Hence *¬*(*P → Q*) *→ P* has *T* .

### Other Connectives

*¬*(*P → Q*) ⇒ *P*

We introduce the connectives NAND, NOR which have useful applications in the design of computers.

**NAND:** The word NAND is a combination of ‘NOT‘ and ‘AND‘ where ‘NOT‘ stands for negation and ‘AND‘ for the conjunction. It is denoted by the symbol *↑*.

If *P* and *Q* are two formulas then

*P ↑ Q* ⇔ *¬*(*P* ∧ *Q*) The connective *↑* has the following equivalence:

*P ↑ P* ⇔ *¬*(*P* ∧ *P* ) ⇔ *¬P* ∨ *¬P* ⇔ *¬P* .

(*P ↑ Q*) *↑* (*P ↑ Q*) ⇔ *¬*(*P ↑ Q*) ⇔ *¬*(*¬*(*P* ∧ *Q*)) ⇔ *P* ∧ *Q*. (*P ↑ P* ) *↑* (*Q ↑ Q*) ⇔ *¬P ↑ ¬Q* ⇔ *¬*(*¬P* ∧ *¬Q*) ⇔ *P* ∨ *Q*.

NAND is Commutative: Let *P* and *Q* be any two statement formulas.

(*P ↑ Q*) ⇔ *¬*(*P* ∧ *Q*)

⇔ *¬*(*Q* ∧ *P* ) ⇔

(*Q ↑ P* )

∴ NAND is commutative.

NAND is not Associative: Let *P* , *Q* and *R* be any three statement formulas. Consider *↑* (*Q ↑ R*) ⇔ *¬*(*P* ∧ (*Q ↑ R*)) ⇔ *¬*(*P* ∧ (*¬*(*Q* ∧ *R*)))

⇔ *¬P* ∨ (*Q* ∧ *R*)) (*P ↑ Q*) *↑ R* ⇔ *¬*(*P* ∧ *Q*) *↑ R*

⇔ *¬*(*¬*(*P* ∧ *Q*) ∧ *R*) ⇔

(*P* ∧ *Q*) ∨ *¬R*

Therefore the connective *↑* is not associative.

**NOR:** The word NOR is a combination of ‘NOT‘ and ‘OR‘ where ‘NOT‘ stands for negation and

‗OR‘ for the disjunction. It is denoted by the symbol *↓*.

If *P* and *Q* are two formulas then

*P ↓ Q* ⇔ *¬*(*P* ∨ *Q*) The connective *↓* has the following equivalence:

*P ↓ P* ⇔ *¬*(*P* ∨ *P* ) ⇔ *¬P* ∧ *¬P* ⇔ *¬P* .

8) Indicate which of the following formulas are well-formed :

1. ¬(p Λ q) b) pΛ¬p V q) c) (pq) Λq d) ¬(p V q)

(*P ↓ Q*) *↓* (*P ↓ Q*) ⇔ *¬*(*P ↓ Q*) ⇔ *¬*(*¬*(*P* ∨ *Q*)) ⇔ *P* ∨ *Q*. (*P ↓ P* ) *↓* (*Q ↓ Q*) ⇔ *¬P ↓ ¬Q* ⇔ *¬*(*¬P* ∨ *¬Q*) ⇔ *P* ∧ *Q*.

NOR is Commutative: Let *P* and *Q* be any two statement formulas.

(*P ↓ Q*) ⇔ *¬*(*P* ∨ *Q*)

⇔ *¬*(*Q* ∨ *P* ) ⇔

(*Q ↓ P* )

∴ NOR is commutative.

NOR is not Associative: Let *P* , *Q* and *R* be any three statement formulas. Consider

*P↓* (*Q ↓ R*) ⇔ *¬*(*P* ∨ (*Q ↓ R*))

⇔ *¬*(*P* ∨ (*¬*(*Q* ∨ *R*)))

⇔ *¬P* ∧ (*Q* ∨ *R*) (*P ↓ Q*) *↓ R* ⇔ *¬*(*P* ∨ *Q*) *↓ R*

⇔ *¬*(*¬*(*P* ∨ *Q*) ∨ *R*) ⇔

(*P* ∨ *Q*) ∧ *¬R*

Therefore the connective *↓* is not associative.

Evidently, *P ↑ Q* and *P ↓ Q* are duals of each other.

Since

*¬*(*P* ∧ *Q*) ⇔ *¬P* ∨ *¬Q*

*¬*(*P* ∨ *Q*) ⇔ *¬P* ∧ *¬Q.*

Example: Express *P ↓ Q* interms of *↑* only. Solution:

*↓ Q* ⇔ *¬*(*P* ∨ *Q*)

⇔ (*P* ∨ *Q*) *↑* (*P* ∨ *Q*)

⇔ [(*P ↑ P* ) *↑* (*Q ↑ Q*)] *↑* [(*P ↑ P* ) *↑* (*Q ↑ Q*)]

Example: Express *P ↑ Q* interms of *↓* only. (May-2012) Solution: *↑ Q* ⇔ *¬*(*P* ∧ *Q*)

⇔ (*P* ∧ *Q*) *↓* (*P* ∧ *Q*)

⇔ [(*P ↓ P* ) *↓* (*Q ↓ Q*)] *↓* [(*P ↓ P* ) *↓* (*Q ↓ Q*)]

### Truth Tables

Example: Show that (*A* ⊕ *B*) ∨ (*A ↓ B*) ⇔ (*A ↑ B*). (May-2012) Solution: We prove this by constructing truth table.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *A* | *B* | *A* ⊕ *B* | *A ↓ B* | (*A* ⊕ *B*) ∨ (*A ↓ B*) | *A ↑ B* |
| T | T | F | F | F | F |
| T | F | T | F | T | T |
| F | T | T | F | T | T |
| F | F | F | T | T | T |

As columns (*A* ⊕ *B*) ∨ (*A ↓ B*) and (*A ↑ B*) are identical.

∴ (*A* ⊕ *B*) ∨ (*A ↓ B*) ⇔ (*A ↑ B*).

**Well formed Formulas:**

1. Any statement variable is a well formed formulas
2. For any WFF A, ¬A is a WFF
3. If A and B are WFFS, then A∧ B , A ∨B, A*→*B, A↔ B and A⊕B are WFF
4. A finite string of symbols is a WFF only when it is constructed by step 1,2,3

**Examples**:

1. Indicate which of the following formulas are well-formed **:**

a) **¬(p Λ q)** b) **pΛ¬p V q)** c) **(p🡪q) 🡪Λq**  d) **¬(p V q)**

Sol:a) WFF b) not WFF c) not WFF d) WFF