**Proposition:**

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

**Examples:**

**Truth value example:**

Islamabad is a capital city of Pakistan. It is a proposition and its truth value is true.

3. 1 + 1 = 22 It is a proposition and its truth value is false.

**Proposition or not examples:**

What time is it? Not a proposition

Read this carefully. Not a proposition.

**Because above these are not declarative sentences.**

x + 1 = 2 . It is not a proposition. It is neither true nor false.

**But:**

Rule: If the sentence is preceded by other sentences that make the pronoun or variable reference clear,

then the sentence is a statement.

**Example:** x > 2

x = 1 It is a statement (Proposition) because x>2 clear the value of “x”.

x > 2 is a statement with truth value of false.(2=1 is false).

**More:**

Example: George Boole was a famous mathematician. He is renowned.

He is renowned is a statement with truth value true.

Example: Bill Gates is an American. He is very rich.

He is very rich is a statement with truth value true.

**There are two types of proposition:**

a) Simple proposition

b) Compound proposition

**Logical connectives:**

There are four important logical connectives--conjunctions, disjunctions, conditional statements, and bi-conditional statements as well as negations.

We can use these connectives to build up complicated compound propositions involving any number of propositional variables.

AND, OR, NOT, if…then (implication or conditionals), if and only if (bi-implications or bi-conditional) are called logical connectives.

**Rule:**

**Note**: No of rows in the truth table depend on the number of propositions.

**No of rows = 2n**

where n = Number of propositions

**Conditional:** p → q, 'if p, then q'

**1st True and 2nd False gives False, rest are True**

**Biconditional:** p ↔ q ≡ (p → q ) ∧ (q → p)

**Both same True, different False**

**Exclusive or:(Negation of biconditional)** p ⨁ q ≡ ¬(p ↔ q)

**Different True, same False**

**Different ways to express conditional statement (implication)**

1. ''if p, then q''

2. ''if p, q ''

3. ''p is sufficient for q''

4. ''q if p''

5. ''q when p''

6. ''a necessary condition for p is q''

7. ''q unless ¬p''

8. ''p implies q''

9. ''p only if q''

10.''a sufficient condition for q is p''

11.''q whenever p''

12.''q is necessary for p''

13.''q follows from p''

**Common ways to express biimplications** :

''p is necessary and sufficient for q ''

''if p then q , and conversely''

''p iff q''

The last way of expressing the biconditional statement p ↔ q uses the abbreviation ''iff'' for ''if and only if.''

Note that p ↔ q has exactly the same truth value as (p → q) ∧ (q → p).

**CONVERSE:** The converse of p → q is the proposition q → p

**CONTRAPOSITIVE:** The contrapositive of p → q is the proposition ¬q → ¬p

**INVERSE:** The contrapositive of p → q is the proposition ¬p → ¬q

**Rule**: **p → q ≡ ¬q → ¬p(Contrapositive)**

**System specifications:**

System specifications should be consistent, that is, they should not contain conflicting requirements that could be used to derive a contradiction.

When specifications are not consistent, there would be no way to develop a system that satisfies all specifications.

**Proposition Equivalence:**

**Tautology:** A compound proposition that is always true, no matter what the truth values of the propositions that occur in it, is called a tautology.

**Contradiction:** A compound proposition that is always false is called a contradiction.

**Contingency:** A compound proposition that is neither a tautology nor a contradiction is called a contingency.

**Logical Equivalences:**

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**.

**Examples:**

¬(p V q) and ¬p Λ ¬q **(DE Morgan’s Law)**

p V (q Λ r) and (p V q) Λ (p V r) **(Distributive property)**

p V (q Λ r) and (p V q) Λ (p V r)



