Rules of Inference in Artificial intelligence

# Inference:

In artificial intelligence, we need intelligent computers which can create new logic from old logic or by evidence, **so generating the conclusions from evidence and facts is termed as Inference**.

# Inference rules:

Inference rules are the templates for generating valid arguments. Inference rules are applied to derive proofs in artificial intelligence, and the proof is a sequence of the conclusion that leads to the desired goal.

In inference rules, the implication among all the connectives plays an important role. Following are some terminologies related to inference rules:

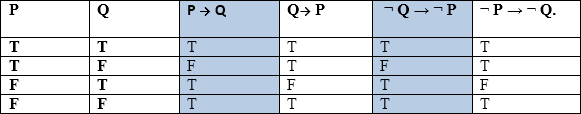
**Implication:** It is one of the logical connectives which can be represented as P → Q. It is a Boolean expression.

**Converse:** The converse of implication, which means the right-hand side proposition goes to the left-hand side and vice-versa. It can be written as Q → P.

**Contrapositive:** The negation of converse is termed as contrapositive, and it can be represented as ¬ Q → ¬ P.

**Inverse:** The negation of implication is called inverse. It can be represented as ¬ P → ¬ Q.

From the above term some of the compound statements are equivalent to each other, which we can prove using truth table:



Hence from the above truth table, we can prove that P → Q is equivalent to ¬ Q → ¬ P, and Q→ P is equivalent to ¬ P → ¬ Q.

# Types of Inference rules:

## Modus Ponens:

The Modus Ponens rule is one of the most important rules of inference, and it states that if P and P → Q is true, then we can infer that Q will be true. It can be represented as:



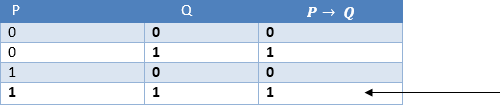
**Example:**

Statement-1: "If I am sleepy then I go to bed" ==> P→ Q Statement-2: "I am sleepy" ==> P

Conclusion: "I go to bed." ==> Q.

Hence, we can say that, if P→ Q is true and P is true then Q will be true.

**Proof by Truth table:**



## Modus Tollens:

The Modus Tollens rule state that if P→ Q is true and **¬ Q is true, then ¬ P** will also true. It can be represented as:

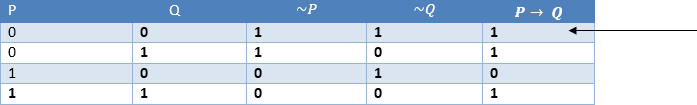


**Statement-1:** "If I am sleepy then I go to bed" ==> P→ Q

**Statement-2:** "I do not go to the bed."==> ~Q

**Statement-3:** Which infers that "**I am not sleepy**" => ~P

**Proof by Truth table:**



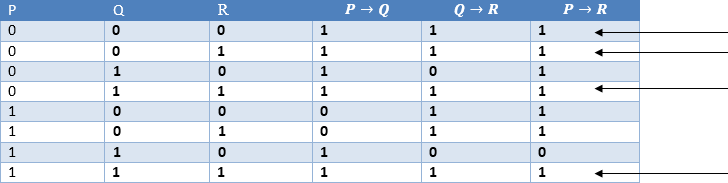
## Hypothetical Syllogism:

The Hypothetical Syllogism rule state that if P→R is true whenever P→Q is true, and Q→R is true. It can be represented as the following notation:

**Example:**

**Statement-1:** If you have my home key then you can unlock my home. **P→Q Statement-2:** If you can unlock my home then you can take my money. **Q→R Conclusion:** If you have my home key then you can take my money. **P→R**

**Proof by truth table:**



## Disjunctive Syllogism:

The Disjunctive syllogism rule state that if P∨Q is true, and ¬P is true, then Q will be true. It can be represented as:



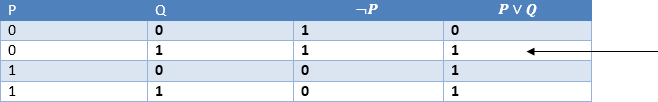
**Example:**

**Statement-1:** Today is Sunday or Monday. ==>P∨Q

**Statement-2:** Today is not Sunday. ==> ¬P

**Conclusion:** Today is Monday. ==> Q

**Proof by truth-table:**



## Addition:

The Addition rule is one the common inference rule, and it states that If P is true, then P∨Q will be true.



**Example:**

**Statement:** I have a vanilla ice-cream. ==> P

**Statement-2:** I have Chocolate ice-cream.

**Conclusion:** I have vanilla or chocolate ice-cream. ==> (P∨Q)

**Proof by Truth-Table:**



## Simplification:

The simplification rule state that if **P**𝖠 **Q** is true, then **Q or P** will also be true. It can be represented as:



**Proof by Truth-Table:**



## Resolution:

The Resolution rule state that if P∨Q and ¬ P𝖠R is true, then Q∨R will also be true. **It can be represented as**



**Proof by Truth-Table:**

