***Unit 3***

***Knowledge Representation***

***Gate Smasher***

* **Major important factor** to **make machine intelligent** is knowledge.
* **Knowledge is facts, skills acquired through education and experience** while **intelligence is ability to use that knowledge.**
* **Reasoning** is **processing of knowledge**.
* It’s required **to put knowledge, intelligence and reasoning in machine** to **make it intelligent/ to make correct decisions.**
* **Store knowledge in the machine** so that **when right time comes machine will be able to use knowledge, and take intelligent decisions.**
* Stored knowledge wrong then wrong interpretations, wrong decisions therefore **proper representation of knowledge is very important**. **Eg: program writing from gate smasher.**

**Different methods of knowledge representation**

1. **Logic:** Propositional logic, Predicate logic
2. **Rules:** Provide rules in terms of if-then as a guideline for machine. **Eg: Fuzzy logic**
3. **Semantic nets:** **Semantic** meaning **simple** and **net** meaning **graph**. Therefore, **semantic net: meaningful graph**. It is graphical way of representation of knowledge. It will be a graph that will include objects, their relationships**. Eg: google graphs, Cow, elephant, tiger**
4. **Frames:** It uses **slots** meaning **objects** and **fillers** meaning **attributes**. Eg object cow and its attributes like four legs, a tail etc. Its more precise than semantic nets. **It has fixed structure like a railway ticket**
5. **Script:** Stronger way of knowledge than frames. It’s **like script of a movie.** Whole script is given to machine.

***Edureka***

**Knowledge representation and reasoning r**epresents **the information from the real world for a computer to understand and then utilize this knowledge to solve complex real-life problems like communicating with human beings in natural language.**

**Store the data and allow machine to learn from it and behave intelligently like a human being.**

Different kinds of knowledge that needs to be represented.

**Objects, events, performance, facts, meta knowledge, knowledge base.**

**Types of knowledge:**

1. **Declarative knowledge:** Concepts, facts and objects expressed in a declarative sentence. Eg: Today extra class for AI-ML is scheduled at 4.00 pm.
2. **Structural knowledge:** Basic problem solving that describes relationship between concept and objects.
3. **Procedural knowledge:** Procedure to do something. It includes, rules strategies etc. Eg: Program writing.
4. **Meta Knowledge:** Knowledge about other types of knowledge. Eg: advanced tutoring system designed to help students learn mathematics. This AI system not only teaches math concepts but also has meta knowledge about its own teaching strategies and the student's learning patterns.
5. **Heuristic knowledge:** Experts knowledge in the field.

***\*\*Javapoint***

What is knowledge representation?

knowledge. Human knows things, which is knowledge and as per their knowledge they perform various actions in the real world. **But how machines do all these things comes under knowledge representation and reasoning**. Hence, we can describe Knowledge representation as following:

* Knowledge representation and reasoning (KR, KRR) is the part of Artificial intelligence which concerned with AI agents thinking and how thinking contributes to intelligent behaviour of agents.
* **It is responsible for representing information about the real world so that a computer can understand and can utilize this knowledge to solve the complex real-world problems such as diagnosis a medical condition or communicating with humans in natural language.**
* It is also a way which describes how we can represent knowledge in artificial intelligence. **Knowledge representation is not just storing data into some database, but it also enables an intelligent machine to learn from that knowledge and experiences so that it can behave intelligently like a human.**

What to Represent:

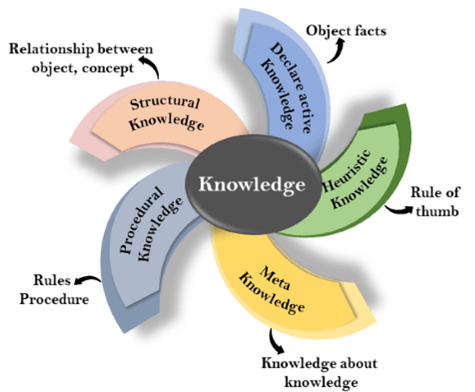
Following are the kind of knowledge which needs to be represented in AI systems:

* **Object:** All the facts about objects in our world domain. **E.g., Guitars contains strings, trumpets are brass instruments.**
* **Events:** Events are the actions which occur in our world.
* **Performance:** It describe behaviour which involves knowledge about **how to do things.**
* **Meta-knowledge:** It is knowledge about what we know.
* **Facts:** Facts are the **truths about the real world** and what we represent.
* **Knowledge-Base: The central component of the knowledge-based agents is the knowledge base. It is represented as KB. The Knowledgebase is a group of the Sentences (Here, sentences are used as a technical term and not identical with the English language).**

**Knowledge:** Knowledge is awareness or familiarity gained by experiences of facts, data, and situations. Following are the types of knowledge in artificial intelligence:

Types of knowledge

Following are the various types of knowledge:



**1. Declarative Knowledge:**

* Declarative knowledge is to know about something.
* It includes **concepts, facts, and objects.**
* It is also called descriptive knowledge and expressed in declarative sentences.
* It is simpler than procedural language.

**2. Procedural Knowledge**

* It is also known as **imperative knowledge**.
* Procedural knowledge is a type of **knowledge which is responsible for knowing how to do something.**
* It can be directly applied to any task.
* It includes **rules, strategies, procedures, agendas**, etc.
* Procedural knowledge depends on the task on which it can be applied.

**3. Meta-knowledge:**

* **Knowledge about the other types of knowledge** is called Meta-knowledge.
* Meta knowledge in AI **refers to knowledge about how knowledge is organized, used, and manipulated within an artificial intelligence system**. It involves understanding and reasoning about the underlying structure, rules, and relationships that govern the system's knowledge representation and processing.

**4. Heuristic knowledge:**

* Heuristic knowledge is representing **knowledge of some experts in a field** or **subject.**
* Heuristic knowledge is **rules of thumb based on previous experiences,** **awareness of approaches, and which are good to work but not guaranteed.**

**5. Structural knowledge:**

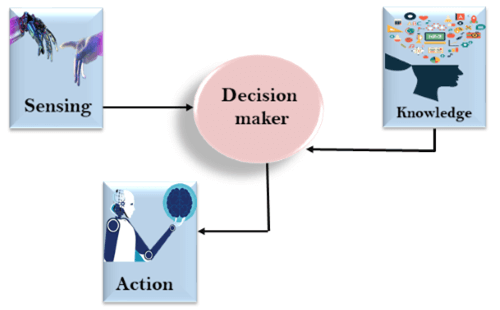
* Structural knowledge is **basic knowledge to problem-solving**.
* It describes relationships between various concepts such as kind of, part of, and grouping of something.
* It describes the **relationship that exists between concepts or objects**.

The relation between knowledge and intelligence:

Knowledge of real-worlds plays a vital role in intelligence and same for creating artificial intelligence. Knowledge plays an important role in demonstrating intelligent behaviour in AI agents. An agent is only able to accurately act on some input when he has some knowledge or experience about that input.

Let's suppose if you met some person who is speaking in a language which you don't know, then how you will able to act on that. The same thing applies to the intelligent behaviour of the agents.

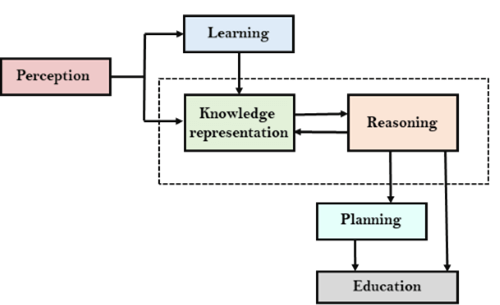
As we can see in below diagram, there is one decision maker which act by sensing the environment and using knowledge. But if the knowledge part will not present then, it cannot display intelligent behaviour.



AI knowledge cycle:

An Artificial intelligence system has the following components for displaying intelligent behaviour:

* Perception
* Learning
* Knowledge Representation and Reasoning
* Planning
* Execution



The above diagram is showing **how an AI system can interact with the real world** and what components help it to show intelligence. AI system has **Perception component by which it retrieves information from its environment.** It can be visual, audio or another form of sensory input. **The learning component is responsible for learning from data captured by Perception component.** In the complete cycle, the main components are **knowledge representation and Reasoning.** These two components are involved in **showing the intelligence in machine-like humans.** These two components are independent with each other but also coupled together. **The planning and execution depend on analysis of Knowledge representation and reasoning.**

Approaches to knowledge representation:

There are **mainly four approaches** to knowledge representation, which are given below

1. Simple relational knowledge:

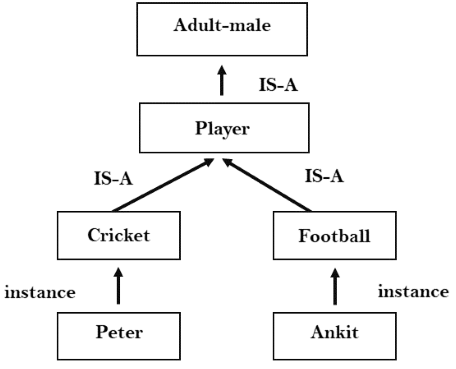
* It is the **simplest way of storing facts** which **uses the relational method**, and each fact about a set of the object is set out systematically in columns.
* This approach of knowledge representation is famous in database systems **where the relationship between different entities is represented**.
* This approach has little opportunity for inference.

**Example: The following is the simple relational knowledge representation.**

|  |  |  |
| --- | --- | --- |
| **Player** | **Weight** | **Age** |
| Player1 | 65 | 23 |
| Player2 | 58 | 18 |
| Player3 | 75 | 24 |

2. Inheritable knowledge:

* In the inheritable knowledge approach, **all data must be stored into a hierarchy of classes.**
* All classes should be arranged in a generalized form or a hierarchal manner.
* In this approach, we apply inheritance property.
* **Elements inherit values from other members of a class**.
* This approach contains inheritable knowledge which shows a relation between instance and class, and it is called instance relation.
* Every individual frame can represent the collection of attributes and its value.
* In this approach, objects and values are represented in Boxed nodes.
* We use **Arrows which point from objects to their values**.
* **Example:**



3. Inferential knowledge:

* Inferential knowledge approach represents knowledge **in the form of formal logics.**
* This approach can be used to derive more facts.
* It guaranteed correctness.
* **Example:** Let's suppose there are two statements:
  1. Marcus is a man
  2. All men are mortal  
     Then it can represent as;  
       
     **man(Marcus)  
     ∀x = man (x) ----------> mortal (x)s**

4. Procedural knowledge:

* Procedural knowledge approach uses small programs and codes which describes how to do specific things, and how to proceed.
* In this approach, one important rule is used which is **If-Then rule**.
* In this knowledge, we can use various coding languages such as **LISP language** and **Prolog language**.
* We can easily represent heuristic or domain-specific knowledge using this approach.
* But it is not necessary that we can represent all cases in this approach.

Requirements for knowledge Representation system:

A good knowledge representation system must possess the following properties.

**1. Representational Accuracy:**  
KR system should have the ability to represent all kind of required knowledge.

**2. Inferential Adequacy:**  
KR system should have ability to manipulate the representational structures to produce new knowledge corresponding to existing structure.

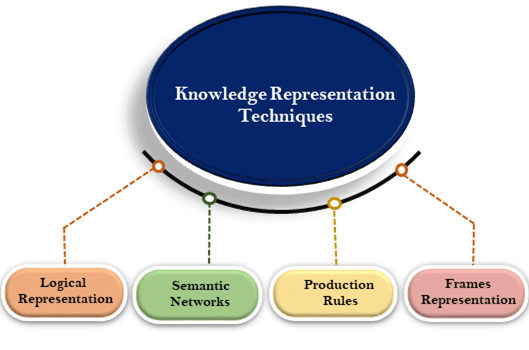
**3. Inferential Efficiency:**  
The ability to direct the inferential knowledge mechanism into the most productive directions by storing appropriate guides.

**4. Acquisitional efficiency-** The ability to acquire the new knowledge easily using automatic methods.

Techniques of knowledge representation

There are mainly four ways of knowledge representation which are given as follows:

1. **Logical Representation**
2. **Semantic Network Representation**
3. **Frame Representation**
4. **Production Rules**



1. Logical Representation

Logical representation is a **language with some concrete rules** which deals with propositions and **has no ambiguity in representation**. Logical representation **means drawing a conclusion based on various conditions**. This **representation lays down some important communication rules.** It consists of **precisely defined syntax and semantics** which supports the sound inference. **Each sentence can be translated into logics using syntax and semantics.**

Syntax:

* Syntaxes are the rules which decide **how we can construct legal sentences** in the logic.
* It **determines which symbol** we can use in knowledge representation.
* How to write those symbols.

Semantics:

* Semantics are the **rules by which we can interpret the sentence** in the logic.
* Semantic also involves **assigning a meaning to each sentence**.

Logical representation can be categorised into mainly two logics:

1. **Propositional Logics**
2. **Predicate logics**

Advantages of logical representation:

1. **Logical representation enables us to do logical reasoning.**
2. **Logical representation is the basis for the programming languages.**

Disadvantages of logical Representation:

1. Logical representations have some restrictions and are challenging to work with.
2. Logical representation technique may not be very natural, and inference may not be so efficient.

Note: Do not be confused with logical representation and logical reasoning as logical representation is a representation language and reasoning is a process of thinking logically.

2. Semantic Network Representation

Semantic networks are alternative of predicate logic for knowledge representation. In Semantic networks, we can represent our knowledge in the form of graphical networks. This network consists of nodes representing objects and arcs which describe the relationship between those objects. Semantic networks can categorize the object in different forms and can also link those objects. Semantic networks are easy to understand and can be easily extended.

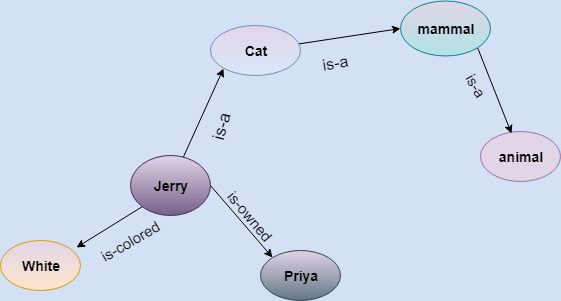
This representation consists of mainly two types of relations:

1. IS-A relation (Inheritance)
2. Kind-of-relation

**Example:** Following are some statements which we need to represent in the form of nodes and arcs.

Statements:

1. Jerry is a cat.
2. Jerry is a mammal
3. Jerry is owned by Priya.
4. Jerry is brown coloured.
5. All Mammals are animal.



In the above diagram, we have represented the different type of knowledge in the form of nodes and arcs. Each object is connected with another object by some relation.

Drawbacks in Semantic representation:

1. Semantic networks take more computational time at runtime as we need to traverse the complete network tree to answer some questions. It might be possible in the worst-case scenario that after traversing the entire tree, we find that the solution does not exist in this network.
2. Semantic networks try to model human-like memory (Which has 1015 neurons and links) to store the information, but in practice, it is not possible to build such a vast semantic network.
3. These types of representations are inadequate as they do not have any equivalent quantifier, e.g., for all, for some, none, etc.
4. Semantic networks do not have any standard definition for the link names.
5. These networks are not intelligent and depend on the creator of the system.

Advantages of Semantic network:

1. Semantic networks are a natural representation of knowledge.
2. Semantic networks convey meaning in a transparent manner.
3. These networks are simple and easily understandable.

3. Frame Representation

A frame is a record like structure which consists of a collection of attributes and its values to describe an entity in the world. Frames are the AI data structure which divides knowledge into substructures by representing stereotypes situations. It consists of a collection of slots and slot values. These slots may be of any type and sizes. Slots have names and values which are called facets.

**Facets:** The various aspects of a slot is known as **Facets**. Facets are features of frames which enable us to put constraints on the frames. Example: IF-NEEDED facts are called when data of any particular slot is needed. A frame may consist of any number of slots, and a slot may include any number of facets and facets may have any number of values. A frame is also known as **slot-filter knowledge representation** in artificial intelligence.

Frames are derived from semantic networks and later evolved into our modern-day classes and objects. A single frame is not much useful. Frames system consist of a collection of frames which are connected. In the frame, knowledge about an object or event can be stored together in the knowledge base. The frame is a type of technology which is widely used in various applications including Natural language processing and machine visions.

Example: 1

Let's take an example of a frame for a book

|  |  |
| --- | --- |
| **Slots** | **Filters** |
| **Title** | Artificial Intelligence |
| **Genre** | Computer Science |
| **Author** | Peter Norvig |
| **Edition** | Third Edition |
| **Year** | 1996 |
| **Page** | 1152 |

Example 2:

Let's suppose we are taking an entity, Peter. Peter is an engineer as a profession, and his age is 25, he lives in city London, and the country is England. So following is the frame representation for this:

|  |  |
| --- | --- |
| **Slots** | **Filter** |
| **Name** | Peter |
| **Profession** | Doctor |
| **Age** | 25 |
| **Marital status** | Single |
| **Weight** | 78 |

Advantages of frame representation:

1. The frame knowledge representation makes the programming easier by grouping the related data.
2. The frame representation is comparably flexible and used by many applications in AI.
3. It is very easy to add slots for new attribute and relations.
4. It is easy to include default data and to search for missing values.
5. Frame representation is easy to understand and visualize.

Disadvantages of frame representation:

1. In frame system inference mechanism is not be easily processed.
2. Inference mechanism cannot be smoothly proceeded by frame representation.
3. Frame representation has a much-generalized approach.

4. Production Rules

Production rules system consist of (**condition, action**) pairs which mean, "If condition then action". It has mainly three parts:

* The set of production rules
* Working Memory
* The recognize-act-cycle

In production rules agent checks for the condition and if the condition exists then production rule fires and corresponding action is carried out. The condition part of the rule determines which rule may be applied to a problem. And the action part carries out the associated problem-solving steps. This complete process is called a recognize-act cycle.

The working memory contains the description of the current state of problems-solving and rule can write knowledge to the working memory. This knowledge match and may fire other rules.

If there is a new situation (state) generates, then multiple production rules will be fired together, this is called conflict set. In this situation, the agent needs to select a rule from these sets, and it is called a conflict resolution.

Example:

* **IF (at bus stop AND bus arrives) THEN action (get into the bus)**
* **IF (on the bus AND paid AND empty seat) THEN action (sit down).**
* **IF (on bus AND unpaid) THEN action (pay charges).**
* **IF (bus arrives at destination) THEN action (get down from the bus).**

Advantages of Production rule:

1. The production rules are expressed in natural language.
2. The production rules are highly modular, so we can easily remove, add or modify an individual rule.

Disadvantages of Production rule:

1. Production rule system does not exhibit any learning capabilities, as it does not store the result of the problem for the future uses.
2. During the execution of the program, many rules may be active hence rule-based production systems are inefficient.

Propositional logic in Artificial intelligence

**Propositional logic** (PL) is the **simplest form of logic** where all the statements are made by propositions. **A proposition is a declarative statement** which is either true or false. It is a technique of knowledge representation in logical and mathematical form.

Example:

a) It is Sunday.

b) The Sun rises from West. (False proposition)

c) 3+3= 7(False proposition)

d) 5 is a prime number.

**Following are some basic facts about propositional logic:**

* Propositional logic is **also called Boolean logic** as it works on 0 and 1.
* In propositional logic, we **use symbolic variables** to represent the logic, and we can use any symbol for a representing a proposition, such A, B, C, P, Q, R, etc.
* Propositions can be **either true or false**, but it cannot be both.
* Propositional logic consists of **an object, relations or function** and **logical connectives**.
* These **connectives** are also called **logical operators.**
* The **propositions a**nd **connectives** are the **basic elements** of the propositional logic.
* **Connectives** can be said as a **logical operator** which connects two sentences.
* A proposition formula **which is always true** is called **tautology**, and it is also called a **valid sentence.**
* A proposition formula **which is always false** is called **Contradiction**.
* A proposition formula which has both true and false values is called **Contingency.**
* Statements which are questions, commands, or opinions are not propositions such as "**Where is Rohini**", "**How are you**", "**What is your name**", are not propositions.

Syntax of propositional logic:

The syntax of propositional logic defines the allowable sentences for the knowledge representation. There are two types of Propositions:

1. **Atomic Propositions**
2. **Compound propositions**

* **Atomic Proposition:** Atomic propositions are the simple propositions. It consists of a single proposition symbol. These are the sentences which must be either true or false.

**Example:** a) 2+2 is 4, it is an atomic proposition as it is a **true** fact.

b) "The Sun is cold" is also a proposition as it is a **false** fact.

* **Compound proposition:** Compound propositions are constructed by combining simpler or atomic propositions, using parenthesis and logical connectives.

**Example:** a) "It is raining today, and street is wet."

b) "Ankit is a doctor, and his clinic is in Mumbai."

Logical Connectives:

Logical connectives are used to connect two simpler propositions or representing a sentence logically. We can create compound propositions with the help of logical connectives. There are mainly five connectives, which are given as follows:

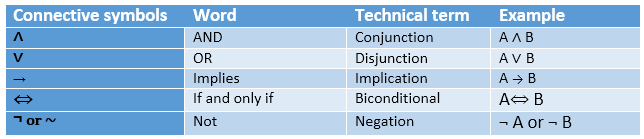
1. **Negation:** A sentence such as ¬ P is called negation of P. A literal can be either Positive literal or negative literal.

**Example:** Today is not Friday.

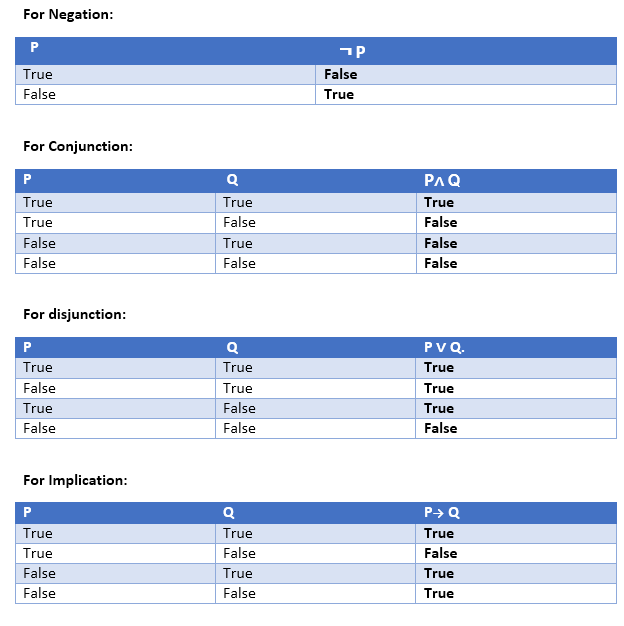
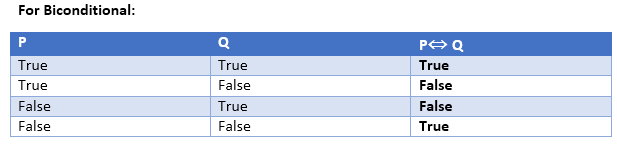
**P**=Today is a Friday**.** Then---Today is not Friday. **→ ¬P**

1. **Conjunction:** A sentence which has **∧**connective such as, **P ∧ Q** is called a conjunction.  
   **Example:** Rohan is intelligent and hardworking. It can be written as,  
   **P= Rohan is intelligent**,  
   **Q= Rohan is hardworking. → P∧ Q**.
2. **Disjunction:** A sentence which has ∨ connective, such as **P ∨ Q**. is called disjunction, where P and Q are the propositions.  
   **Example: "Ritika is a doctor or Engineer"**,  
   **Here P= Ritika is Doctor. Q= Ritika is Engineer, so we can write it as  P ∨ Q.**
3. **Implication:** A sentence such as P → Q, is called an implication. Implications are also known as if-then rules. It can be represented as  
               **If** it is raining, then the street is wet.  
         **Let P= It is raining, and Q= Street is wet, so it is represented as P → Q**
4. **Biconditional:** A sentence such as **P⇔ Q is a Biconditional sentence, example If I am breathing, then I am alive**  
   **P= I am breathing, Q= I am alive, it can be represented as P ⇔ Q.**

Following is the summarized table for Propositional Logic Connectives:

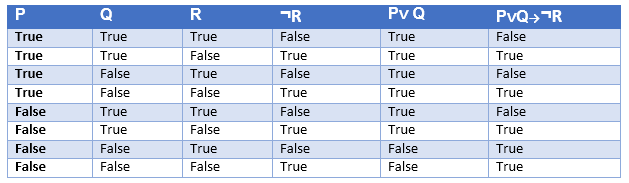


Truth Table:

In propositional logic, we need to know the truth values of propositions in all possible scenarios. We can combine all the possible combination with logical connectives, and the representation of these combinations in a tabular format is called **Truth table**. Following are the truth table for all logical connectives: 

Truth table with three propositions:

We can build a proposition composing three propositions P, Q, and R. This truth table is made-up of 8n Tuples as we have taken three proposition symbols.



Precedence of connectives:

Just like arithmetic operators, there is a precedence order for propositional connectors or logical operators. This order should be followed while evaluating a propositional problem. Following is the list of the precedence order for operators:

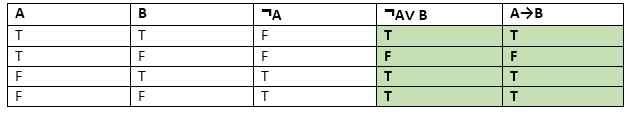
|  |  |
| --- | --- |
| **Precedence** | **Operators** |
| First Precedence | Parenthesis |
| Second Precedence | Negation |
| Third Precedence | Conjunction (AND) |
| Fourth Precedence | Disjunction (OR) |
| Fifth Precedence | Implication |
| Six Precedence | Biconditional |

Note: For better understanding use parenthesis to make sure of the correct interpretations. Such as ¬R∨ Q, It can be interpreted as (¬R) ∨ Q.

Logical equivalence:

Logical equivalence is one of the features of propositional logic. Two propositions are said to be logically equivalent if and only if the columns in the truth table are identical to each other.

Let's take two propositions A and B, so for logical equivalence, we can write it as A⇔B. In below truth table we can see that column for ¬A∨ B and A→B, are identical hence A is Equivalent to B



Properties of Operators:

* **Commutativity:**
  + P∧ Q= Q ∧ P, or
  + P ∨ Q = Q ∨ P.
* **Associativity:**
  + (P ∧ Q) ∧ R= P ∧ (Q ∧ R),
  + (P ∨ Q) ∨ R= P ∨ (Q ∨ R)
* **Identity element:**
  + P ∧ True = P
  + P ∨ True= True.
* **Distributive:**
  + P∧ (Q ∨ R) = (P ∧ Q) ∨ (P ∧ R).
  + P ∨ (Q ∧ R) = (P ∨ Q) ∧ (P ∨ R).
* **DE Morgan's Law:**
  + ¬ (P **∧** Q) = (¬P) **∨** (¬Q)
  + ¬ (P **∨** Q) = (¬ P) **∧** (¬Q).
* **Double-negation elimination:**
  + ¬ (¬P) = P.

Limitations of Propositional logic:

* We cannot represent relations like ALL, some, or none with propositional logic. Example:
  1. **All the girls are intelligent.**
  2. **Some apples are sweet.**
* Propositional logic **has limited expressive power.**
* In propositional logic, we **cannot describe statements** in terms of their properties or **logical relationships**.

Predicate/First Order in Artificial intelligence

It is extended part of propositional logic that articulates the natural languages in a easy way.

It models the world in terms of objects.

Properties of objects help in differentiating from one object to other.

Eg: object Teacher Student

Property ID1 Roll No.

The symbols are used to represent the objects, properties and their relations.

Different symbols used in FOL are: A-Z, 0-9, underscore.

There are two types of Quantifiers used to define scope of the objects.

|  |  |
| --- | --- |
| **Universal** | **Existential** |
| **Eg: All boys like football.**  **∀x: Boys(x) → like(x, football)**  For all x and if x is a boy, then he likes football. | **Eg: Some boys like football.**  **∃x: Boys(x)** ∧ **like(x, football)**  There exist some boys and if x is a boy, then he likes football. |
| It uses **implication** operator **(→)**. | * It uses **conjunction** operator **(∧).** |
| If **x** is a variable, then **∀x** is read as:   * **For all x** * **For each x** * **For every x.** | * If **x** is a variable, then **∃x** is read as **There exists a 'x.'** * **For some 'x.'** * **For at least one 'x.'** |

Some Examples of FOL using quantifier:

**1. All birds fly.**  
In this question the predicate is "**fly(bird)**."  
And since there are all birds who fly so it will be represented as follows.  
              **∀x bird(x) →fly(x)**.

**2. Every man respects his parent.**  
In this question, the predicate is "**respect(x, y)," where x=man, and y= parent**.  
Since there is every man so will use ∀, and it will be represented as follows:  
              **∀x man(x) → respects (x, parent)**.

**3. Some boys play cricket.**  
In this question, the predicate is "**play(x, y)**," where x= boys, and y= game. Since there are some boys so we will use **∃, and it will be represented as**:  
              **∃x boys(x) → play(x, cricket)**.

**4. Not all students like both Mathematics and Science.**  
In this question, the predicate is "**like(x, y)," where x= student, and y= subject**.  
Since there are not all students, so we will use **∀ with negation, so** following representation for this:  
              **¬∀ (x) [ student(x) → like(x, Mathematics) ∧ like(x, Science)].**

**5. Only one student failed in Mathematics.**  
In this question, the predicate is "**failed(x, y)," where x= student, and y= subject**.  
Since there is only one student who failed in Mathematics, so we will use following representation for this:  
              **∃(x) [ student(x) → failed (x, Mathematics) ∧∀ (y) [¬(x==y) ∧ student(y) → ¬failed (x, Mathematics)]**.

Resolution in FOL

Resolution

**It’s a technique by which a theorem can be proved using proof by contradiction.**

Resolution is a theorem proving technique that proceeds by building refutation proofs, i.e., proofs by contradictions. It was invented by a Mathematician John Alan Robinson in the year 1965.

**Resolution is used, if there are various statements are given, and we need to prove a conclusion of those statements.** **Unification is a key concept in proofs by resolutions.** Resolution is a single inference rule which can efficiently operate on the **conjunctive normal form or clausal form**.

**Clause**: Disjunction of literals (an atomic sentence) is called a **clause**. It is also known as a unit clause.

**Conjunctive Normal Form**: A sentence represented as a conjunction of clauses is said to be **conjunctive normal form** or **CNF**.

Steps for Resolution:

1. Conversion of facts into first-order logic.
2. Convert FOL statements into CNF
3. Negate the statement which needs to prove (proof by contradiction)
4. Draw resolution graph (unification).

Steps in Conversion of FOL into CNF

In First order logic resolution, it is required to convert the FOL into CNF as CNF form makes easier for resolution proofs.

* **Eliminate all implication (→) and rewrite**

**∀x man(x) → drink (x, coffee) becomes ∀x ¬** **man(x) V drink (x, coffee)**

* **Move negation (¬)inwards and rewrite**
  + 1. **¬ ¬ P becomes P**
    2. **¬(P V Q) becomes ¬P Λ ¬Q**
    3. **¬(P Λ Q) becomes ¬P V ¬Q**
* **Rename variables or standardize variables**

**Eg: a. All boys like cricket. b. All girls like pink.**

**∀x: boy(x) → like(x, cricket) ∀x: girl(x) → like(x, pink)**

**∀x: ¬boy(x) V like(x, cricket) ∀x: ¬girl(x) V like(x, pink)**

**∀x: ¬boy(x) V like(x, cricket) ∀y: ¬girl (y) V like(y, pink)**

* **Eliminate existential instantiation quantifier by elimination.**  
  In this step, we will eliminate existential quantifier ∃, and this process is known as **Skolemization**.
* **Drop Universal quantifiers.**   
  In this step we will drop all universal quantifier since all the statements are not implicitly quantified so we don't need it.

**∀x: ¬boy(x) V like(x, cricket) ∀y: ¬girl (y) V like(y, pink)**

**¬boy(x) V like(x, cricket) ¬girl (y) V like(y, pink)**

* **Distribute conjunction ∧ over disjunction ¬.  
   (P V Q) V R** **becomes (P V R) V (Q V R)**

**(P Λ Q) V R** **becomes (P V R) Λ (Q V R)**

Unification

Goal of unification: To make the expressions look identical using process of substitution.

It is **matching procedure** that **compares two literals** and **discovers if there exist a set of substitution that makes them identical.**

