# Unit-4 - Knowledge Representation

Knowledge Representation -Knowledge based agents – The Wumpus world – Propositional Logic - syntax, semantics and knowledge base building - inferences – reasoning patterns in propositional logic – predicate logic – representing facts in logic: Syntax and semantics – Unification – Unification Algorithm - Knowledge representation using rules - Knowledge representation using semantic nets - Knowledge representation using frames inferences - Uncertain Knowledge and reasoning Methods.

## Knowledge Representation

Knowledge Representation (KR) is a fundamental concept in Artificial Intelligence (AI) that deals with representing knowledge in a format that can be understood and used by machines.

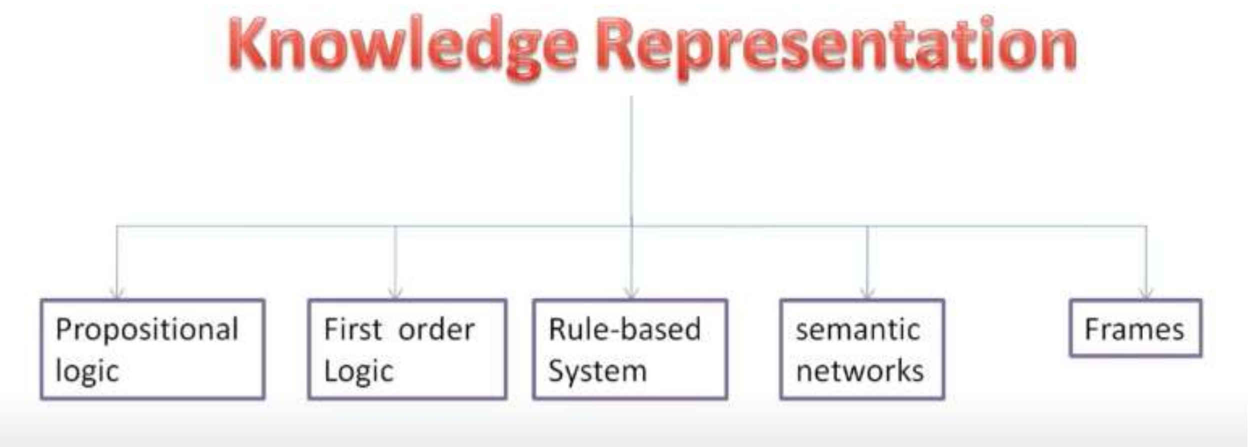
### Types of Knowledge Representation:

**1. Symbolic representation:** Uses symbols and rules to represent knowledge.

**2. Connectionist representation:** Uses artificial neural networks to represent knowledge.

**3. Hybrid representation:** Combines symbolic and connectionist representations.

## Knowledge Representation Techniques:



### 1. Propositional Logic:

- A formal system for representing and reasoning about propositions (statements that can be true or false).

- Uses logical operators (AND, OR, NOT) to combine propositions.

- Example: $(P \land Q) \rightarrow R$ (If P and Q are true, then R is true)

### 2. First-Order Logic (FOL):

- An extension of propositional logic that allows for quantification over objects and relations.

- Uses predicates, variables, and quantifiers ($\forall$, $\exists$) to represent knowledge.

- Example: $\forall x\ (\text{Person}(x) \rightarrow \text{Mortal}(x))$ (All persons are mortal)

### 3. Rule-Based Systems:

- A knowledge representation technique that uses IF-THEN rules to represent knowledge.

- Rules consist of antecedents (conditions) and consequents (actions or conclusions).

- Example: IF temperature > 30°C THEN turn on air conditioner

### 4. Semantic Networks:

- A graphical representation of knowledge that shows relationships between concepts.

- Nodes represent concepts, and edges represent relationships (e.g., IS-A, PART-OF).

- Example: A semantic network representing the relationship between a car, its engine, and its wheels.

### 5. Frames:

- A knowledge representation technique that uses structured frameworks to represent knowledge.

- Frames consist of slots (attributes) and fillers (values).

- Example: A frame representing a person might have slots for name, age, occupation, and address.

### Applications:

1. Expert systems: KR is used to build expert systems that mimic human decision-making.

2. Natural language processing: KR is used to represent and understand human language.

3. Knowledge graphs: KR is used to represent knowledge as a graph of interconnected entities.

### Benefits:

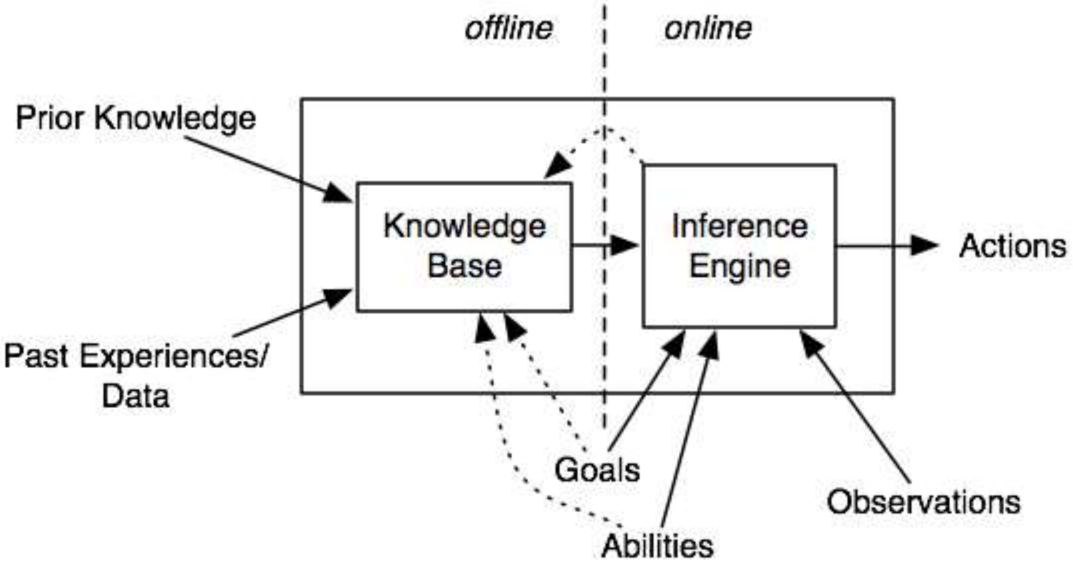
1. Improved decision-making: KR enables machines to make informed decisions.

2. Knowledge sharing: KR facilitates knowledge sharing and reuse.

3. Intelligent systems: KR is a key component of intelligent systems that can reason and learn.

## Knowledge Based Agent

A knowledge-based agent is a type of intelligent agent that uses a knowledge base to make decisions and take actions:



### Components:

**1. Knowledge base:** A repository of knowledge that the agent uses to make decisions.

**2. Inference engine:** A mechanism that uses the knowledge base to draw conclusions and make decisions.

**3. Action selection:** The agent selects actions based on the conclusions drawn from the knowledge base.

## Characteristics:

**1. Knowledge-driven:** The agent's behavior is determined by the knowledge in its knowledge base.

**2. Reasoning:** The agent uses reasoning mechanisms to draw conclusions from its knowledge base.

**3. Decision-making:** The agent makes decisions based on the conclusions drawn from its knowledge base.

## Types:

**1. Expert systems:** Knowledge-based agents that mimic the decision-making abilities of a human expert.

**2. Rule-based systems:** Knowledge-based agents that use IF-THEN rules to make decisions.

## Applications:

**1. Decision support systems:** Knowledge-based agents can provide decision support in various domains.

**2. Expert systems:** Knowledge-based agents can be used to build expert systems that mimic human expertise.

**3. Intelligent systems:** Knowledge-based agents can be used to build intelligent systems that can reason and make decisions.

## Benefits:

**1. Improved decision-making:** Knowledge-based agents can make informed decisions based on a large knowledge base.

**2. Consistency:** Knowledge-based agents can provide consistent decisions and actions.

**3. Scalability:** Knowledge-based agents can handle large amounts of knowledge and scale to meet the needs of complex applications.

## The Wumpus World:

The Wumpus World is a classic problem in artificial intelligence, introduced by Gregory Yob in 1973. It's a simple, grid-based world where an agent must navigate and avoid dangers to achieve a goal.

### Components:

1. Agent: The agent navigates the grid, trying to find gold while avoiding dangers.

2. Wumpus: A monster that can kill the agent if they're in the same location.

3. Pits: Locations that can trap and kill the agent.

4. Gold: The agent's goal is to find and retrieve the gold.

### Agent's Task:

The agent must use reasoning, problem-solving, and decision-making to:

1. Avoid Wumpus and pits: Use sensors to detect dangers and navigate safely.

2. Find gold: Use reasoning to locate the gold and retrieve it.

### Applications:

The Wumpus World is used to:

1. Teach AI concepts: Introduce students to AI problem-solving, reasoning, and decision-making.

2. Test AI algorithms: Evaluate the performance of AI algorithms in a controlled environment.

### Benefits:

1. Simple yet challenging: The Wumpus World provides a simple yet challenging environment for AI research and education.

2. Illustrates AI concepts: The Wumpus World illustrates key AI concepts, such as reasoning, problem-solving, and decision-making.

## PEAS for the Wumpus World:

PEAS stands for Performance measure, Environment, Actuators, and Sensors.

PEAS Components:

**1. Performance measure:**

- Find gold
- Avoid Wumpus and pits
- Minimize steps

**2. Environment:**

- Grid-based world
- Wumpus, pits, and gold locations
- Agent's current location

**3. Actuators:**

- Move forward
- Turn left
- Turn right
- Grab gold

**4. Sensors:**

- Breeze (detect pits)
- Stench (detect Wumpus)
- Glitter (detect gold)
- Bump (detect walls)

## Propositional Logic:

Propositional logic is a branch of logic that deals with statements that can be either true or false.

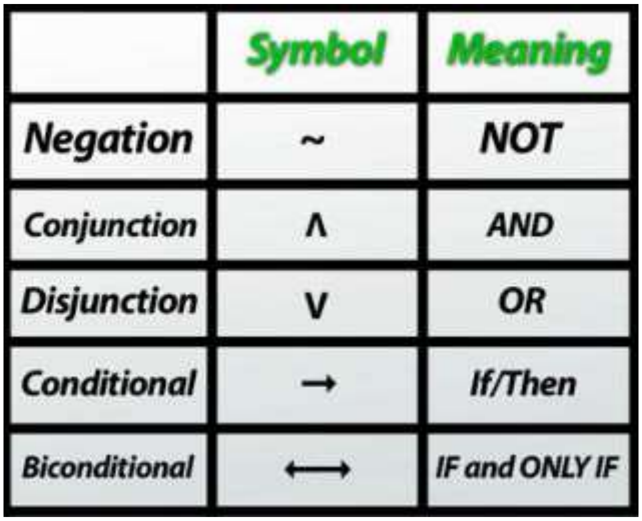
Propositional logic is used for solving complex problems using simple statements. These statements can either be true or false but cannot be both at same time. These propositions form knowledge representation, reasoning and decision-making in AI systems. In this article we will see the basics of propositional logic and its applications in AI.

## Key Concepts:

**1. Propositions:** Statements that can be true or false.

**2. Logical operators:** Used to combine propositions, such as:

- ∧ (conjunction, AND)
- ∨ (disjunction, OR)
- ¬ (negation, NOT)
- → (implication)
- (equivalence)



| | Symbol | Meaning |
|---|---|---|
| Negation | ~ | NOT |
| Conjunction | ∧ | AND |
| Disjunction | V | OR |
| Conditional | → | If/Then |
| Biconditional | ⟷ | IF and ONLY IF |

**Applications:**

1. Artificial intelligence: Propositional logic is used in AI for knowledge representation and reasoning.

2. Computer science: Propositional logic is used in programming, software engineering, and formal verification.

3. Mathematics: Propositional logic is used in mathematical proofs and theorem proving.

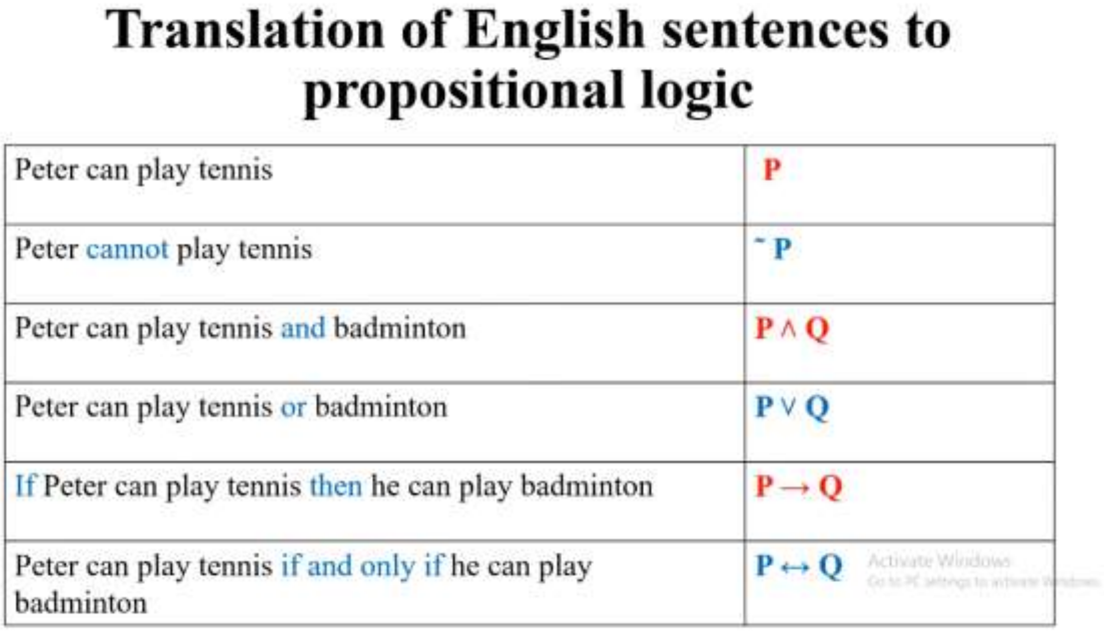
**Benefits:**

1. Formal reasoning: Propositional logic provides a formal framework for reasoning about statements.

2. Clear semantics: Propositional logic has clear and well-defined semantics.

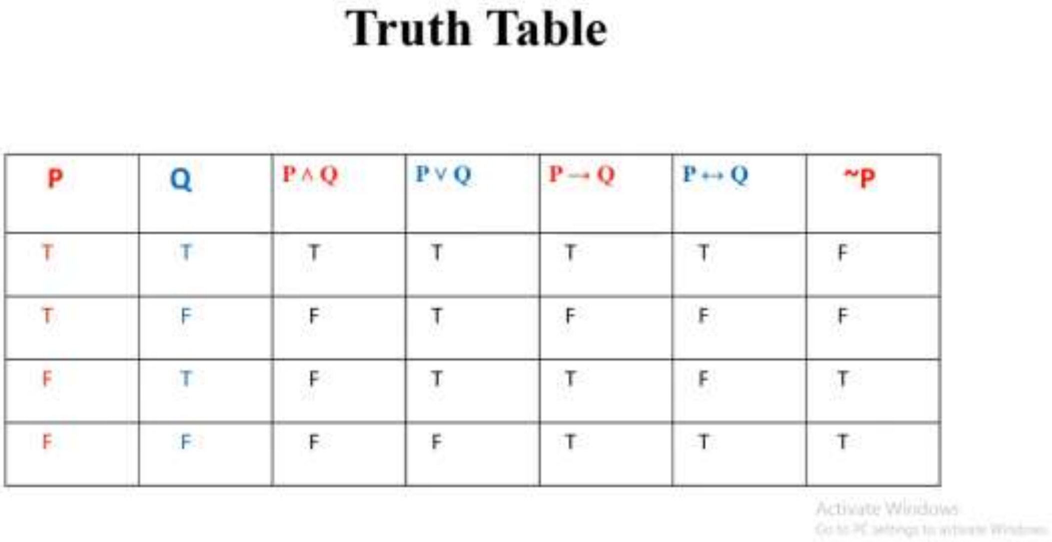
3. Wide applicability: Propositional logic has applications in many fields, including AI, computer science, and mathematics.

**Example**



## Translation of English sentences to propositional logic

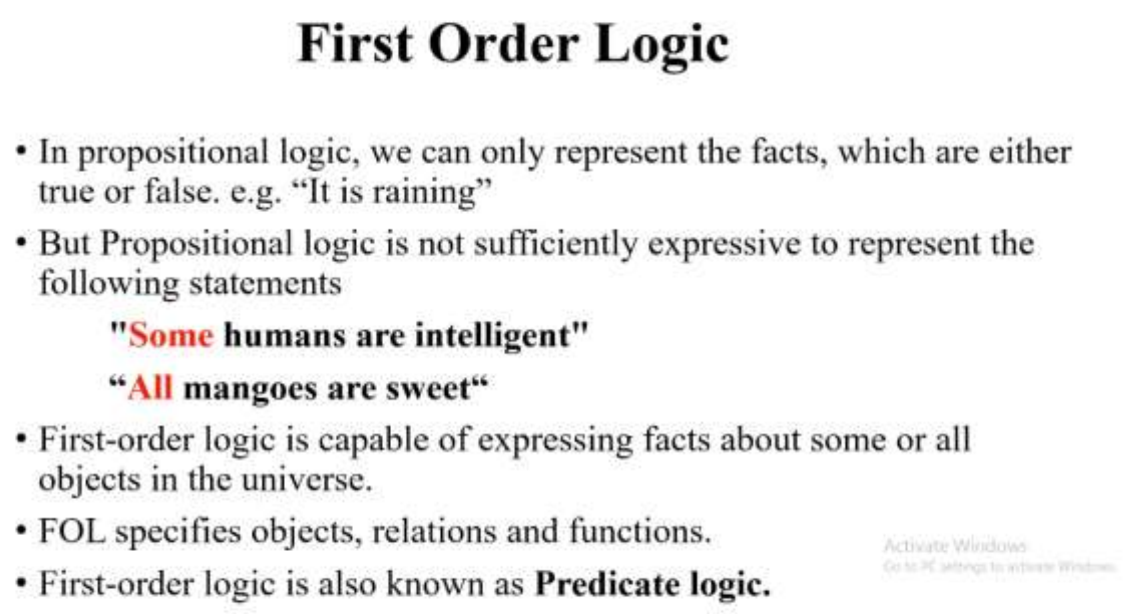
| | |
|---|---|
| Peter can play tennis | P |
| Peter cannot play tennis | ~ P |
| Peter can play tennis and badminton | P ∧ Q |
| Peter can play tennis or badminton | P ∨ Q |
| If Peter can play tennis then he can play badminton | P → Q |
| Peter can play tennis if and only if he can play badminton | P ↔ Q |



## Truth Table

| P | Q | P ∧ Q | P ∨ Q | P → Q | P ↔ Q | ~P |
|---|---|---|---|---|---|---|
| T | T | T | T | T | T | F |
| T | F | F | T | F | F | F |
| F | T | F | T | T | F | T |
| F | F | F | F | T | T | T |

## First Order Logic

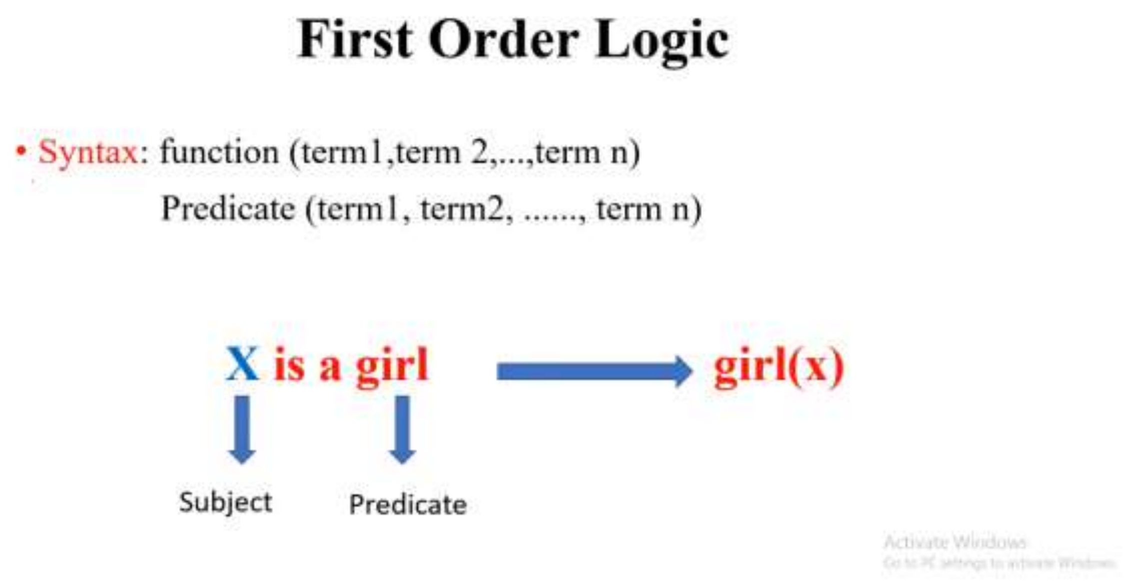


### First Order Logic

- In propositional logic, we can only represent the facts, which are either true or false. e.g. “It is raining”
- But Propositional logic is not sufficiently expressive to represent the following statements

  **"Some humans are intelligent"**

  **“All mangoes are sweet“**
- First-order logic is capable of expressing facts about some or all objects in the universe.
- FOL specifies objects, relations and functions.
- First-order logic is also known as **Predicate logic.**



### First Order Logic

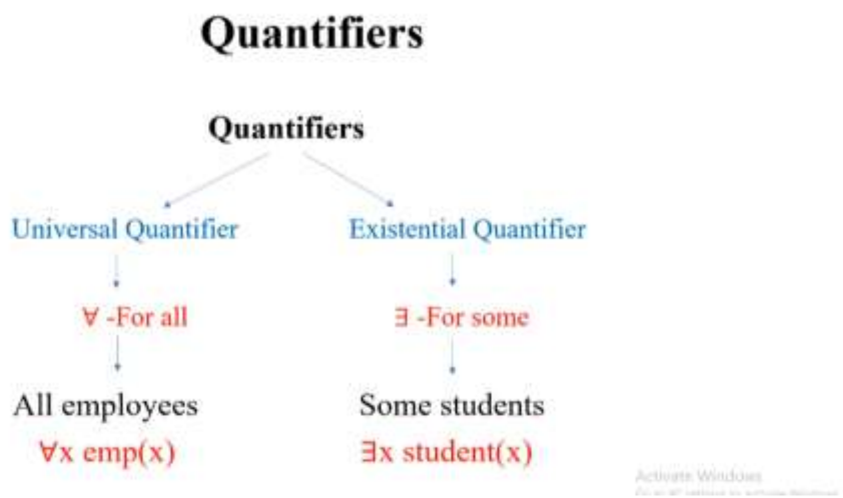
- Syntax: function (term1,term 2,...,term n)

  Predicate (term1, term2, ......, term n)

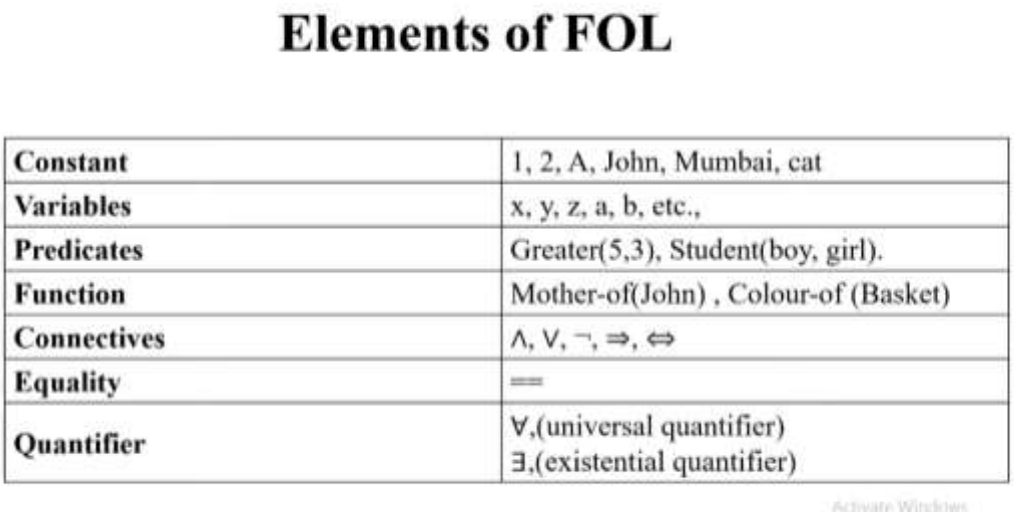
**X is a girl** → **girl(x)**

Subject

Predicate

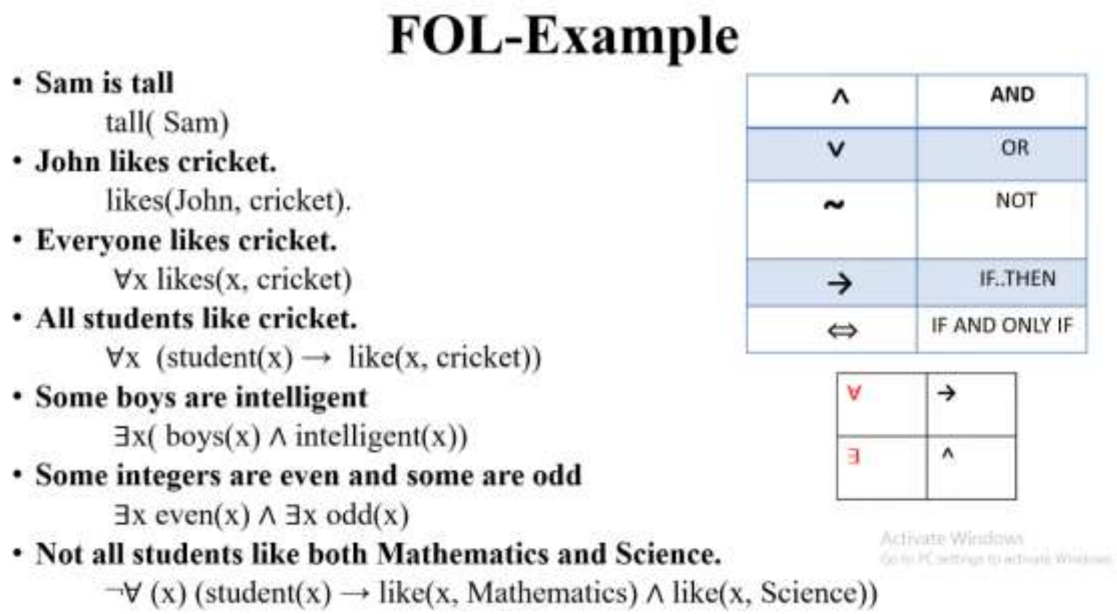


# Quantifiers



# Elements of FOL

| | |
|---|---|
| **Constant** | 1, 2, A, John, Mumbai, cat |
| **Variables** | x, y, z, a, b, etc., |
| **Predicates** | Greater(5,3), Student(boy, girl). |
| **Function** | Mother-of(John) , Colour-of (Basket) |
| **Connectives** | ∧, ∨, ¬, ⇒, ⇔ |
| **Equality** | == |
| **Quantifier** | ∀,(universal quantifier)<br>∃,(existential quantifier) |



# FOL-Example

- **Sam is tall**

  tall( Sam)
- **John likes cricket.**

  likes(John, cricket).
- **Everyone likes cricket.**

  ∀x likes(x, cricket)
- **All students like cricket.**

  ∀x (student(x) → like(x, cricket))
- **Some boys are intelligent**

  ∃x( boys(x) ∧ intelligent(x))
- **Some integers are even and some are odd**

  ∃x even(x) ∧ ∃x odd(x)
- **Not all students like both Mathematics and Science.**

  ¬∀ (x) (student(x) → like(x, Mathematics) ∧ like(x, Science))

| | |
|---|---|
| ∧ | AND |
| ∨ | OR |
| ~ | NOT |
| → | IF..THEN |
| ⇔ | IF AND ONLY IF |

| | |
|---|---|
| ∀ | → |
| ∃ | ∧ |

## Unification

Unification is the process of finding a substitution that makes two logical expressions identical. It's a fundamental concept in artificial intelligence, logic programming, and automated reasoning.

## Unification Algorithm

The Unification Algorithm is a step-by-step procedure for finding a unifier (substitution) that makes two expressions identical.

### Steps:

1. Compare terms: Compare the two terms to be unified.
2. Check for variables: If one term is a variable, bind it to the other term.
3. Check for constants: If both terms are constants, they must be identical.
4. Check for functions: If both terms are functions, unify their arguments recursively.
5. Apply substitution: Apply the substitution to the terms and recursively unify.

### Properties:

1. Soundness: The unification algorithm is sound, meaning that if it finds a unifier, it is correct.
2. Completeness: The unification algorithm is complete, meaning that if a unifier exists, it will be found.

### Example:

Suppose we have two expressions:

1. P(X, Y)
2. P(a, b)

We want to unify these expressions.

### Unification Steps:

1. Compare the predicate symbols: P matches P.
2. Compare the arguments:
    - X is a variable, so it can be bound to a.
    - Y is a variable, so it can be bound to b.

### Resulting Substitution:

The unifier is {X/a, Y/b}, which means:

- X is replaced by a

- Y is replaced by b

Applying this substitution to the first expression P(X, Y) results in P(a, b), which is identical to the second expression.

### Unification Algorithm Outcome:

The two expressions P(X, Y) and P(a, b) are unified, and the resulting substitution is {X/a, Y/b}.

### Applications:

**1. Logic programming:** Unification is used in logic programming languages like Prolog.

**2. Automated reasoning:** Unification is used in automated reasoning systems.

**3. Natural language processing:** Unification can be used in NLP to resolve ambiguities.

## Knowledge Representation using Rules:

Knowledge representation using rules is a way to encode knowledge in a format that can be used by machines:

### Rule Structure:

**1. Antecedent (IF):** The conditions that must be met for the rule to fire.

**2. Consequent (THEN):** The action or conclusion that is drawn when the rule fires.

### Types of Rules:

**1. Deductive rules:** Used for logical deductions.

**2. Production rules:** Used for decision-making and problem-solving.

A rule-based system consists of a set of rules, each in the form of:

IF (conditions) THEN (conclusion)

### Examples:

**Example 1:** Simple Rule

IF (temperature > 30°C) THEN (turn on air conditioner)

**Example 2:** Rule with Multiple Conditions

IF (humidity > 60% AND temperature > 25°C) THEN (start dehumidifier)

**Example 3:** Rule with Complex Conditions

IF (patient has fever AND patient has headache AND patient has sore throat) THEN (patient may have flu)

### Benefits:

1. Easy to understand: Rules are easy to understand and interpret.

2. Flexible: Rules can be added or modified easily.

3. Scalable: Rule-based systems can handle large amounts of knowledge.

### Applications:

1. Expert systems: Rule-based systems are used to build expert systems that mimic human decision-making.

2. Decision support systems: Rule-based systems are used to provide decision support in various domains.

3. Business rule management: Rule-based systems are used to manage business rules and policies.

## Knowledge Representation using Semantic networks

Semantic networks are a powerful tool in the field of artificial intelligence (AI), used to represent knowledge and understand relationships between different concepts. They are graphical representations that connect nodes (representing concepts) with edges (representing relationships). Semantic networks are widely used in natural language processing (NLP), knowledge representation, and reasoning systems.

### Components of Semantic Networks

Semantic networks are made up of several key components:

**1. Lexical Components**

**Nodes:** The fundamental units of a semantic network, representing concepts, entities, or objects within the domain of knowledge. Examples include "Dog," "Animal," or "Tree."

**Labels:** Descriptive names or identifiers associated with the nodes, providing a way to refer to the concepts they represent.

**2. Structural Components**

**Edges/Links:** The connections between nodes, representing relationships such as "is a," "part of," "causes," or "associated with."

**Types of Relationships:** These can include hierarchical relationships (e.g., "is a"), associative relationships (e.g., "related to"), and functional relationships (e.g., "causes" or "results in").

**3. Semantic Components**

**Meanings of Nodes:** The specific meanings or interpretations of the nodes within the context of the network.

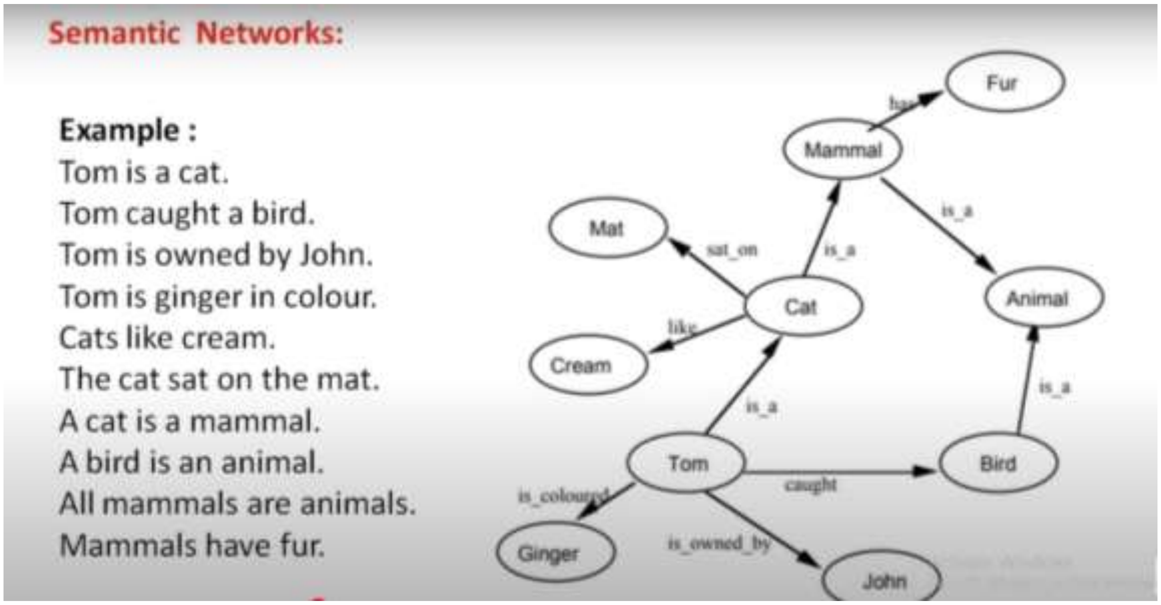
**Interpretation of Relationships:** The understanding of what the edges or links between nodes signify in real-world terms, ensuring the relationships are meaningful and accurately reflect the domain.

**4. Procedural Part**

**Inference Rules:** Rules that allow the network to derive new knowledge from existing relationships. For example, if "Dog is a Mammal" and "Mammal is an Animal," the network can infer that "Dog is an Animal."

**Query Mechanisms:** Procedures for retrieving information from the network based on specific queries or criteria.

**Update Mechanisms:** Rules and processes for adding, modifying, or removing nodes and links as new information is introduced.



Semantic Networks:

**Example :**
Tom is a cat.
Tom caught a bird.
Tom is owned by John.
Tom is ginger in colour.
Cats like cream.
The cat sat on the mat.
A cat is a mammal.
A bird is an animal.
All mammals are animals.
Mammals have fur.

## Knowledge Representation Using Frame

In Artificial Intelligence (AI), frames represent a pivotal concept that helps machines understand and interpret complex real-world scenarios. Originating from cognitive science and knowledge representation, frames are utilized to structure information in a way that allows AI systems to reason, infer, and make decisions.

### Frame Structure:

**1. Frame name:** Identifies the concept or object being represented.

**2. Slots:** Represent attributes or properties of the concept or object.

**3. Fillers:** Values or descriptions that fill the slots.

### Benefits:

**1. Structured representation:** Frames provide a structured and organized way to represent knowledge.

**2. Efficient retrieval:** Frames enable efficient retrieval of knowledge.

**3. Flexible:** Frames can be used to represent complex and nuanced knowledge.

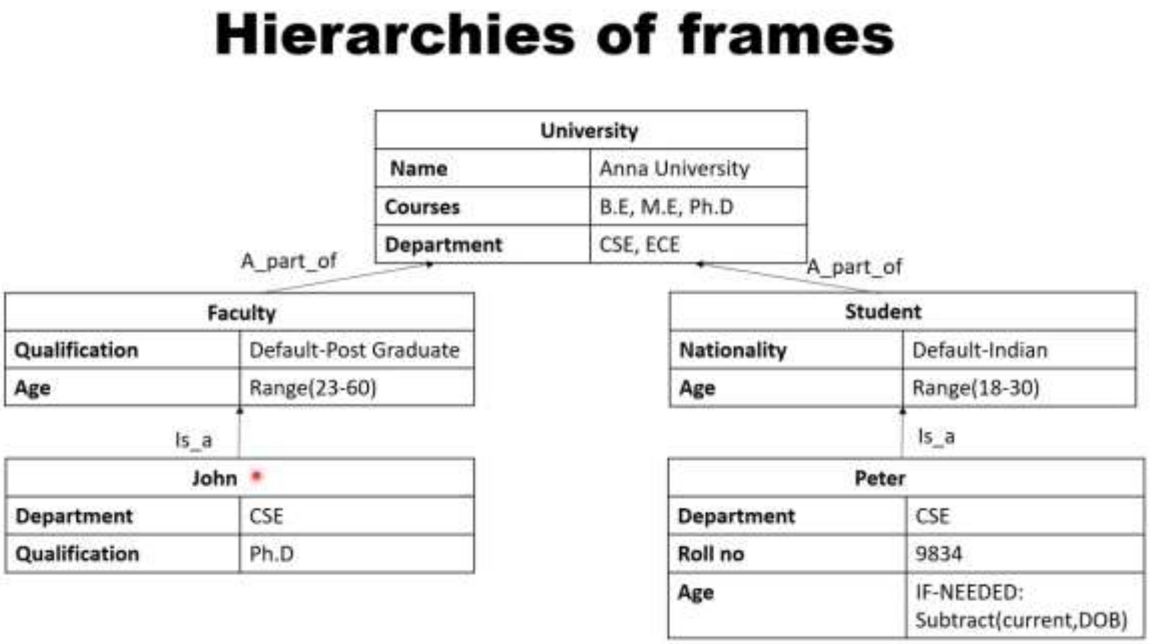
## Applications:

**1. Expert systems:** Frames are used in expert systems to represent domain knowledge.

**2. Knowledge engineering:** Frames are used in knowledge engineering to represent and organize knowledge.

**3. Artificial intelligence:** Frames are used in AI to represent and reason about complex concepts and objects.

## Example



Hierarchies of frames

| University | |
|---|---|
| Name | Anna University |
| Courses | B.E, M.E, Ph.D |
| Department | CSE, ECE |

| Faculty | |
|---|---|
| Qualification | Default-Post Graduate |
| Age | Range(23-60) |

| Student | |
|---|---|
| Nationality | Default-Indian |
| Age | Range(18-30) |

| John | |
|---|---|
| Department | CSE |
| Qualification | Ph.D |

| Peter | |
|---|---|
| Department | CSE |
| Roll no | 9834 |
| Age | IF-NEEDED:<br>Subtract(current,DOB) |

## Uncertain Knowledge

Uncertain Knowledge in Artificial Intelligence (AI) refers to the representation and reasoning of knowledge that is incomplete, imprecise, or unreliable. AI systems often encounter uncertainty due to:

### Sources of Uncertainty:

**1. Noisy or incomplete data:** Data may be incomplete, inaccurate, or noisy.

**2. Ambiguity:** Words, phrases, or concepts can have multiple meanings.

**3. Limited knowledge:** AI systems may not have complete knowledge about a domain.

**Handling Uncertainty in AI:**

**1. Probabilistic models:** Use probability theory to represent uncertainty.

**2. Fuzzy logic:** Use fuzzy sets and rules to represent uncertainty.

**3. Bayesian networks:** Use probabilistic graphical models to represent uncertainty.

**4. Machine learning:** Use machine learning algorithms to learn from uncertain data.

### Applications:

1. Decision-making: AI systems make decisions under uncertainty.

2. Natural language processing: AI systems handle ambiguity and uncertainty in language.

3. Computer vision: AI systems handle uncertainty in image and video analysis.

### Challenges:

1. Representing uncertainty: Accurately representing uncertainty is challenging.

2. Reasoning under uncertainty: Drawing conclusions from uncertain knowledge is complex.

3. Managing uncertainty: Effectively managing uncertainty is essential in AI systems.

### Types of Uncertainty:

**1. Probabilistic uncertainty:** Deals with randomness and chance events.

**2. Fuzzy uncertainty:** Deals with vagueness and imprecision.

### Methods for Handling Uncertainty:

**1. Probability theory:** Uses probability distributions to represent uncertainty.

**2. Fuzzy logic:** Uses fuzzy sets and fuzzy rules to represent uncertainty.

**3. Bayesian networks:** Uses probabilistic graphical models to represent uncertainty.

**4. Dempster-Shafer theory:** Uses belief functions to represent uncertainty.

## Reasoning Methods

Reasoning Methods in Artificial Intelligence (AI) refer to the techniques used to draw conclusions, make decisions, or infer new knowledge from existing information. Some common reasoning methods in AI include:

### Types of Reasoning:

**1. Deductive reasoning:** Uses logical rules to arrive at a certain conclusion.

**2. Inductive reasoning:** Uses specific instances to make generalizations.

**3. Abductive reasoning:** Uses incomplete information to make educated guesses.

### Reasoning Techniques:

**1. Rule-based reasoning:** Uses IF-THEN rules to reason about knowledge.

**2. Case-based reasoning:** Uses past experiences to reason about new situations.

**3. Model-based reasoning:** Uses models of the world to reason about complex systems.