AI ENDSEM UNIT – 4 PYQ

➢ MAY / JUN 2022

Q3)

a) List the inference rules used in prepositional logic? Explain them in detail with suitable example. [9]

Inference rules in **propositional logic** are used to derive **new propositions** from given ones using **logical reasoning**. They are fundamental in theorem proving and logical agents.

Inference Rules in Propositional Logic:

1. **Modus Ponens (Law of Detachment):**
    - **Rule:** If P ⇒ Q and P is true, then Q must be true.
    - **Example:**
        - If *"It is raining ⇒ Roads are wet"*, and *"It is raining"*, then conclude *"Roads are wet"*.
2. **Modus Tollens:**
    - **Rule:** If P ⇒ Q and ¬Q is true, then ¬P is true.
    - **Example:**
        - If *"Alarm is set ⇒ Light is on"*, and *"Light is not on"*, then conclude *"Alarm is not set"*.
3. **And Introduction (Conjunction):**
    - **Rule:** From P and Q, infer P ∧ Q.
    - **Example:**
        - If *"It is cold"* and *"It is windy"*, conclude *"It is cold ∧ windy"*.
4. **And Elimination:**
    - **Rule:** From P ∧ Q, infer either P or Q.
    - **Example:**
        - From *"It is raining ∧ cloudy"*, we can infer *"It is raining"* or *"It is cloudy"*.
5. **Or Introduction (Disjunction):**
    - **Rule:** From P, infer P ∨ Q.
    - **Example:**
        - From *"It is hot"*, we can infer *"It is hot ∨ snowing"*.

6. **Double Negation:**
   - **Rule:** ¬(¬P) is logically equivalent to P.
   - **Example:**
     - If *"¬(¬It is sunny)"*, then conclude *"It is sunny"*.
7. **Resolution Rule:**
   - **Rule:** From P ∨ Q and ¬P ∨ R, infer Q ∨ R.
   - **Example:**
     - From *"It is cold ∨ raining"* and *"¬cold ∨ cloudy"*, infer *"raining ∨ cloudy"*.

b) Explain syntax and semantics of First Order Logic in detail. [8]

First-Order Logic (FOL), also known as predicate logic, extends propositional logic by allowing **quantifiers**, **variables**, and **relations**. It provides a powerful framework to represent real-world knowledge in AI.

Syntax of FOL:

Syntax defines the **formal structure** of valid expressions (also called well-formed formulas).

Components of FOL Syntax:

- **Constants**: Refer to specific objects.
  Example: John, 5, India
- **Variables**: Represent generic entities.
  Example: x, y, z
- **Predicates**: Describe properties or relationships.
  Example: Student(x), Loves(John, Mary)
- **Functions**: Map objects to other objects.
  Example: Father(John) returns John's father.
- **Quantifiers**:
  - **Universal (∀)**: "for all" — ∀x Student(x)
  - **Existential (∃)**: "there exists" — ∃x Loves(x, Mary)
- **Connectives**:
  - ¬ (NOT), ∧ (AND), ∨ (OR), ⇒ (IMPLIES), ⇔ (IFF)

3. Semantics of FOL:

Semantics defines the **meaning** or **interpretation** of FOL sentences in a domain.

Key Concepts in Semantics:

- **Domain (D)**: The set of all possible objects under consideration.
  Example: All people in a university.
- **Interpretation (I)**:
  - Assigns meaning to constants, functions, and predicates.
  - Example: Student(John) is **true** if John is in the set of students.
- **Truth Value**:
  - A sentence in FOL is **true** or **false** based on the interpretation.

Example:

Let the domain be {John, Mary}.
Predicate: Loves(x, Mary)

- Interpretation: John loves Mary → Loves(John, Mary) = true
- Semantics: ∃x Loves(x, Mary) is **true** because at least one such x exists.

The **syntax** of FOL helps in **structuring logical expressions**, while **semantics** gives them **meaning** in a specific domain. Together, they allow AI systems to represent and reason about complex knowledge more expressively than propositional logic.

Q4)

a) Detail the algorithm for deciding entailment in prepositional logic. [8]

In **propositional logic**, **entailment** (written as KB ⊨ α) means that a knowledge base (KB) logically implies the sentence α. That is, α is true in **all models** where KB is true.

To decide entailment, two main methods are used:

- **Model Checking** (Truth Table Method)
- **Inference-Based** (Resolution, Forward/Backward Chaining)

2. Truth Table Algorithm for Entailment:

This is a **model-checking** approach. It checks whether in **all models** where KB is true, α is also true.

Algorithm:

1. **Input:**
   - Knowledge Base KB (a set of propositional sentences)
   - Query sentence α

2. **Extract Symbols:**
   - Collect all propositional symbols from KB and α.
3. **Generate Models:**
   - Generate all possible truth assignments (2^n combinations for n symbols).
4. **Evaluate Models:**
   For each model m:
   - If KB is **true** in model m:
     - Check if α is also **true** in m.
     - If not, return **False** (entailment does not hold).
5. **Return Result:**
   - If α is true in **all models** where KB is true → return **True** (i.e., KB ⊨ α)
   - Else → **False**

3. Example:

Let

- `KB = {P ⇒ Q, P}`
- `α = Q`

Symbols: P, Q

Models:

P	Q	`P ⇒ Q`	`KB true?`	`α true?`
T	T	T	T	T
T	F	F	F	F
F	T	T	F	T
F	F	T	F	F

In the **only row where KB is true** (row 1), `α` is also true. So, `KB ⊨ α`.

The truth table method is **complete and sound** but becomes **computationally expensive** for large numbers of symbols. Therefore, for larger knowledge bases, **efficient inference algorithms** like **resolution** are used in practice.

b) Explain knowledge representation structure and compare them. [9]

Knowledge representation (KR) is a way to organize and store knowledge so that an AI system can reason and make decisions. Various structures are used to represent different types of knowledge.

Knowledge Representation Structures:

1. Logical Representation

- Based on propositional and first-order logic.
- Uses rules, predicates, and quantifiers.
- Example: $\forall x\ (Bird(x) \Rightarrow CanFly(x))$

2. Semantic Network

- Represents knowledge as a **graph** of nodes (objects/concepts) and edges (relations).

- Example: A “Canary” node connected to “Bird” via “is-a”.

3. Frame-Based Representation

- Data structures for representing a **stereotyped situation**.
- Uses **frames** (objects) and **slots** (attributes).
- Example:

```yaml
Frame: Dog
  Type: Animal
  Legs: 4
  Sound: Bark
```

4. Production Rules

- Represent knowledge in **if-then** format.
- Useful in **rule-based systems** or **expert systems**.
- Example:
  IF sky is cloudy THEN it might rain

5. Ontologies

- A formal representation of a **set of concepts** and relationships.
- Used in knowledge graphs and semantic web.

STRUCTURE	ADVANTAGES	DISADVANTAGES	USE CASES
LOGIC	Precise and expressive	Hard to scale; not intuitive	Automated theorem proving
SEMANTIC NETWORK	Intuitive, visual, easy to understand	Ambiguous; lacks formal semantics	Concept mapping, NLP
FRAME-BASED	Structured, supports inheritance	Limited reasoning capability	Object-oriented AI systems
PRODUCTION RULES	Simple, modular, easy to update	Difficult to manage large rule sets	Expert systems (e.g., MYCIN)
ONTOLOGY	Standardized, web-friendly, scalable	Requires domain expertise to build	Semantic web, intelligent

	agents

➢ NOV / DEC 2022

Q3)

a) Explain Wumpus world environment giving its PEAS description. [9]

1.Wumpus World Environment Overview:

- The Wumpus World is a simple, imaginary 4x4 grid-based environment used to demonstrate knowledge-based agents.
- The agent must navigate the grid, avoid hazards, and find gold.
- Hazards include:
  - **Wumpus** (a monster) — deadly if encountered.
  - **Pits** — agent dies if it falls into one.
- The goal is to grab the gold and exit safely.

2. Percepts in Wumpus World:

The agent has limited sensing:

- **Stench** – Wumpus is in an adjacent cell.
- **Breeze** – There's a pit in an adjacent cell.
- **Glitter** – Gold is in the current cell.
- **Bump** – Agent bumped into a wall.
- **Scream** – Wumpus has been killed.

Component	Description
P (Performance)	+Gold collected, –Falling into pit, –Eaten by Wumpus, –Using arrow unnecessarily, +Safe return
E (Environment)	4x4 grid of rooms with hazards (Wumpus, pits), gold, and walls
A (Actuators)	Agent's movement (left, right, forward), shoot arrow, grab, climb

S (Sensors)	Percepts: Stench, Breeze, Glitter, Bump, Scream

Wumpus World is a classic example of a **partially observable**, **non-deterministic**, **sequential** environment.
It demonstrates how agents can use **logical inference** and **knowledge-based reasoning** to make safe decisions.

b) Explain different inference rules in FOL with suitable example [8]

In **First-Order Logic (FOL)**, inference rules help derive new knowledge from known facts and rules. The most common inference rules include:

1. Universal Instantiation (UI):
If a statement is true for **all elements**, then it is true for **any specific instance**.

- **Rule:**
  From ∀x P(x), we can infer P(a)
- **Example:**
  ∀x Human(x) ⇒ Mortal(x)
  ⇒ Human(Socrates) ⇒ Mortal(Socrates)

2. Existential Instantiation (EI):
If a property holds for **some element**, introduce a new constant representing it.

- **Rule:**
  From ∃x P(x), we infer P(c) (c is a new constant)
- **Example:**
  ∃x Bird(x)
  ⇒ Bird(Tweety) (assuming Tweety is a bird)

3. Generalized Modus Ponens:
A generalized form of modus ponens for FOL.

- **Rule:**
  From P(x) ⇒ Q(x) and P(a), infer Q(a)
- **Example:**
  Human(x) ⇒ Mortal(x) and Human(Aristotle)
  ⇒ Mortal(Aristotle)

4. Resolution Rule (for FOL):
Used in **automated theorem proving**, combines two clauses to eliminate a literal.

- **Rule:**
  From P(x) ∨ A and ¬P(y) ∨ B, infer A ∨ B (after unification)

- **Example:**
  Clause 1: Animal(x) ∨ Likes(x, Fish)
  Clause 2: ¬Animal(Dog)
  ⇒ Likes(Dog, Fish)

5. Unification:
Not an inference rule, but a **mechanism** used to make two predicates match by finding a **substitution**.

- **Example:**
  To unify Loves(x, y) and Loves(John, Mary),
  ⇒ substitution: {x/John, y/Mary}

These inference rules in FOL enable reasoning over generalized statements using **instantiation**, **modification**, and **logical resolution**. They are crucial for **knowledge-based systems** and **automated reasoning**.

Q4)
a) Write an propositional logic for the statement [10]
i) "All birds fly"
ii) "Every man respect his parents"

➔ Don't know !!!

b) Differentiate between propositional logic and First order logic. [7]

Aspect	Propositional Logic (PL)	First-Order Logic (FOL)
1. Basic Unit	Propositions (e.g., P, Q, R)	Predicates with arguments (e.g., Loves(x, y), Human(x))
2. Expressiveness	Less expressive; cannot express relationships or quantifiers	More expressive; can represent objects, properties, and relations
3. Quantifiers	Not supported	Supports universal (∀) and existential (∃) quantifiers
4. Variables	No variables	Uses variables (x, y, etc.)
5. Domain Knowledge	Can express only simple true/false facts	Can express facts about objects and their relationships
6. Example	P → Q (If P is true then Q is true)	∀x (Human(x) → Mortal(x)) (All humans are mortal)
7. Use Case	Simple rule-based systems, truth tables	Complex AI systems, knowledge bases, expert systems

- **MAY / JUN 2023**

Q3

**a) What is an Agent? Name any 5 agents around you. Explain Knowledge-Based Agent with Wumpus World.
List and explain in short the various steps of knowledge engineering process.
[9 Marks]**

What is an Agent?

An **Agent** is anything that can perceive its environment through sensors and act upon that environment through actuators to achieve specific goals.

Five Agents Around You

1. Smartphone assistant (e.g., Siri, Google Assistant)
2. Autonomous vacuum cleaner (e.g., Roomba)
3. Self-driving car
4. Thermostat (e.g., Nest)
5. Chatbot on a website

Knowledge-Based Agent with Wumpus World

- **Knowledge-Based Agent**:
  An agent that uses a **knowledge base** to store facts about the world and applies **inference rules** to derive new knowledge and decide on actions.
- **Wumpus World Example:**
  The Wumpus World is a grid-based environment where an agent must navigate safely, avoid dangers (like the Wumpus monster and pits), and find gold.

The agent uses **percepts** (like breeze or stench) to infer facts (e.g., breeze indicates a nearby pit). The agent's knowledge base contains these facts and rules, allowing it to make safe moves by logical inference rather than trial and error.

Steps of Knowledge Engineering Process

1. **Knowledge Acquisition**
   Collecting domain-specific facts and rules from experts or data.

2. **Knowledge Representation**
   Structuring the acquired knowledge in a formal way (e.g., logic, semantic networks).
3. **Knowledge Validation and Verification**
   Ensuring the knowledge is accurate, consistent, and complete.
4. **Knowledge Implementation**
   Encoding the knowledge into a knowledge base and integrating it with an inference engine.
5. **Knowledge Maintenance**
   Updating and refining the knowledge base as new information becomes available.

b) Consider the following axioms:

- **If a triangle is isosceles, then its two sides AB and AC are equal,**
- **If AB and AC are equal, then angles B and C are equal,**
- **ABC is an equilateral triangle,**

(i) Represent these facts in predicate logic.
(ii) Explain Inference in Propositional Logic.
[9 Marks]

Part (i) — Representation in Predicate Logic:

Let's define some predicates:

- IsIsosceles(T) — Triangle T is isosceles
- EqualSides(T, AB, AC) — Sides AB and AC of triangle T are equal
- EqualAngles(T, B, C) — Angles B and C of triangle T are equal
- IsEquilateral(T) — Triangle T is equilateral

Let the triangle be ABC.

---

1. **If a triangle is isosceles, then sides AB and AC are equal:**

∀T (IsIsosceles (T) ⇒ EqualSides(T,AB,AC))

2. **If sides AB and AC are equal, then angles B and C are equal:**

∀T (EqualSides (T,AB,AC) ⇒EqualAngles(T,B,C)

ABC is an equilateral triangle:

IsEquilateral(ABC)

Part (ii) — Explanation of Inference in Propositional Logic:

Inference in propositional logic is the process of deriving new propositions (conclusions) from a set of given propositions (premises) using valid logical rules. The goal is to find whether a proposition logically follows from others.

Key points about inference:

- It is based on inference rules like Modus Ponens, Modus Tollens, Resolution, etc.
- If the premises are true, then the conclusion inferred by applying inference rules is also guaranteed to be true.
- Inference is used in automated theorem proving, knowledge-based systems, and AI reasoning.

Example of inference:

Given:

- $P \Rightarrow Q$ (If P then Q)
- P is true

Using **Modus Ponens**, we infer:

- Q is true

Summary:

- Inference allows logical reasoning from known facts to new conclusions.
- It is essential for knowledge-based agents to make decisions based on existing knowledge.

Q4) a) Write the following sentences in FOL (any 2) (using types of quantifiers). [9]

i) Every number is either negative or has a square root .
ii) Every connected and circuit-free graph is a tree .
iii) Some people are either religious or pious .
iv) There is a barber who shaves all men in the town who do not shave themselves.

Let's translate any two sentences into FOL:

1. Every number is either negative or has a square root.

- Let the domain be numbers.
- Predicates:
  - `Negative(x)` means "x is negative"
  - `HasSquareRoot(x)` means "x has a square root"

FOL representation:

$$\forall x \left(Negative(x) \lor HasSquareRoot(x) \right)$$

---

2. Every connected and circuit-free graph is a tree.

- Domain: Graphs
- Predicates:
  - `Connected(g)` means "graph g is connected"
  - `CircuitFree(g)` means "graph g has no circuits"
  - `Tree(g)` means "graph g is a tree"

FOL representation:

$$\forall g \left(Connected(g) \land CircuitFree(g) \Rightarrow Tree(g) \right)$$

b) What is Resolution? Solve the following statement by using resolution algorithm. Draw suitable resolution graph. . [9]

i) Rajesh like all kind of food.

ii) Apple and vegetables are food.

iii) Anything anyone eats and is not killed is food.

iv) Ajay eats peanuts and still alive.

Prove that Rajesh like bananas.

Part 1: What is Resolution?

Resolution is a rule of inference used in propositional and first-order logic for automated theorem proving. It works by refutation:

- To prove a statement, add its negation to the knowledge base.
- Use the resolution rule to derive a contradiction (an empty clause).
- If a contradiction is found, the original statement is true.

Resolution combines clauses containing complementary literals to infer new clauses, gradually reducing the problem until contradiction or no new clauses can be inferred.

Part 2: Given Statements

1. Rajesh likes all kinds of food.
2. Apple and vegetables are food.
3. Anything anyone eats and is not killed by is food.
4. Ajay eats peanuts and is still alive.

Goal: Prove that Rajesh likes bananas.

Step 1: Define predicates and constants

- Like(Rajesh, x) — Rajesh likes x
- Food(x) — x is food
- Apple, Vegetables, Bananas, Peanuts — constants
- Eats(Ajay, x) — Ajay eats x
- KilledBy(x) — x kills Ajay (or generally anyone)

Step 2: Convert statements into FOL and then to clauses

1. **Rajesh likes all food:**

$$\forall x (Food(x) \Rightarrow Like(Rajesh, x))$$

   Clause form:

$$\neg Food(x) \lor Like(Rajesh, x)$$

2. **Apple and vegetables are food:**

$$Food(Apple)$$

$$Food(Vegetables)$$

3. **Anything eaten and not killed by is food:**

$$\forall x((Eats(Ajay, x) \land \neg KilledBy(x)) \Rightarrow Food(x))$$

   Equivalent to:

$$\forall x(\neg Eats(Ajay, x) \lor KilledBy(x) \lor Food(x))$$

4. **Ajay eats peanuts and is alive (not killed):**

$$Eats(Ajay, Peanuts)$$

$$\neg KilledBy(Peanuts)$$

Step 3: Negate the goal to prove and add to KB

Goal: Rajesh likes bananas

$$Like(Rajesh, Bananas)$$

Negate:

$$\neg Like(Rajesh, Bananas)$$

Step 4: List all clauses (in CNF form)

- $\neg Food(x) \lor Like(Rajesh, x)$
- $Food(Apple)$
- $Food(Vegetables)$
- $\neg Eats(Ajay, x) \lor KilledBy(x) \lor Food(x)$
- $Eats(Ajay, Peanuts)$
- $\neg KilledBy(Peanuts)$

Step 5.2: We have $\neg Like(Rajesh, Bananas)$ (negated goal), so resolve with above:

Resolving:

$$(\neg Food(Bananas) \vee Like(Rajesh, Bananas)), \quad \neg Like(Rajesh, Bananas)$$

Gives:

$$\neg Food(Bananas)$$

Step 5.3: To contradict $\neg Food(Bananas)$, show $Food(Bananas)$ is true.

Using clause (4):

$$\neg Eats(Ajay, Bananas) \vee KilledBy(Bananas) \vee Food(Bananas)$$

If we can prove $Eats(Ajay, Bananas)$ and $\neg KilledBy(Bananas)$, then $Food(Bananas)$ holds.

But we have only:

- $Eats(Ajay, Peanuts)$ (not bananas)
- $\neg KilledBy(Peanuts)$

No direct information about $Bananas$.

Step 5.4: Since no direct info about bananas, assume Rajesh likes all food (given in clause 1), so if bananas are food, he likes bananas.

But bananas' food status is unknown. Since no info about Ajay eating bananas or whether bananas kill, we cannot prove Food(Bananas)Food(Bananas)Food(Bananas).

The resolution proof cannot conclude that Rajesh likes bananas given the current knowledge base because the status of bananas as food is unknown.

ALTERNATIVE ANSWER !!!

b) What is Resolution?

Resolution is a rule of inference used in logic to prove statements by contradiction. It works by combining clauses with complementary literals to produce new clauses, aiming to derive an empty clause (contradiction), which confirms the original statement is true.

Problem Statement

- Rajesh likes all kinds of food.
  $\forall x(Food(x) \Rightarrow Like(Rajesh, x))$
- Apple and vegetables are food.
  $Food(Apple)$, $Food(Vegetables)$
- Anything eaten and not killing is food.
  $\forall x((Eats(Ajay, x) \land \neg KilledBy(x)) \Rightarrow Food(x))$
- Ajay eats peanuts and is alive.
  $Eats(Ajay, Peanuts)$, $\neg KilledBy(Peanuts)$
- **Prove:** Rajesh likes bananas.

Step 1: Convert to clauses

- $\neg Food(x) \vee Like(Rajesh, x)$
- $Food(Apple)$
- $Food(Vegetables)$
- $\neg Eats(Ajay, x) \vee KilledBy(x) \vee Food(x)$
- $Eats(Ajay, Peanuts)$
- $\neg KilledBy(Peanuts)$
- $\neg Like(Rajesh, Bananas)$ (negated goal)

Step 2: Resolution

- From clause 1 with $x = Bananas$:
  $\neg Food(Bananas) \vee Like(Rajesh, Bananas)$
- Resolve with negated goal:
  $\neg Like(Rajesh, Bananas)$
- Result:
  $\neg Food(Bananas)$

Step 3: Show $Food(Bananas)$ (to get contradiction)

- Using clause 4 for $x = Bananas$:
  $\neg Eats(Ajay, Bananas) \vee KilledBy(Bananas) \vee Food(Bananas)$
- No info about Ajay eating bananas or bananas killing anyone. Cannot prove $Food(Bananas)$.

Conclusion

Since Food(Bananas)Food(Bananas)Food(Bananas) can't be proven, resolution cannot conclude Like(Rajesh,Bananas)Like(Rajesh, Bananas)Like(Rajesh,Bananas) with the given knowledge.

- **NOV / DEC 2023**

Q3)

a) What is an Agent. Name any 5 agents around you Explain Knowledge

based agent with Wumpus World. List and explain in short the various

steps of knowledge engineering process [9]

- **ALREADY DONE !!**

b) If a triangle is isosceles, then its two sides AB and AC are equal. If AB and AC are equal, then angle B and C are equal. ABC is an equilateral triangle. Represent these facts in predicate logic [9]

- **DONE ALREADY !!**

Q4)

a) Write the following sentences in FOL(using types of quantifiers) [9]

i) All birds fly

ii)Some boys play cricket

iii)A first cousin is a child of a parent's sibling

iv)You can fool all the people some of the time and some of the people all the time, but you cannot fool all the people all the time

- **YET NOT DONE !!**

b) What is Knowledge Representation using propositional Logic? Compare

propositional and predicate Logic. [9]

Knowledge Representation is the process of encoding information about the world into a form that a computer system can utilize to solve complex tasks like reasoning and decision-making.

Propositional Logic (also called Boolean logic) represents knowledge using **propositions** which are statements that are either true or false. These propositions are combined using logical connectives like AND ($\wedge$), OR ($\vee$), NOT ($\neg$), IMPLIES ($\Rightarrow$), and BICONDITIONAL ($\Leftrightarrow$).

- Each proposition is an atomic statement, e.g., "It is raining" represented as P.
- Complex statements are built by connecting propositions, e.g., P⇒Q means "If it is raining, then the roads are wet".
- Propositional logic can express facts and simple rules and is useful for reasoning about truth values.

Example:

- Let P = "It is raining"
- Let Q =The ground is wet"
- Rule: If it is raining, then the ground is wet
  P⇒Q

Aspect	**Propositional Logic**	**Predicate (First-Order) Logic**
Expressiveness	Deals with whole propositions as atomic units	Deals with objects, their properties, and relations between them
Structure	Simple statements (propositions)	Includes quantifiers, predicates, functions, variables
Quantifiers	No quantifiers (cannot express "all" or "some")	Supports universal (∀) and existential (∃) quantifiers
Representation	Cannot represent internal structure of objects or relations	Can represent complex relationships like "All humans are mortal"
Example	P = "It is raining"	Human(x)⇒Mortal(x), where x is a variable
Reasoning power	Limited to truth of fixed statements	More powerful, allows reasoning about objects and their properties
Application	Suitable for simple decision problems	Suitable for knowledge-rich domains like natural language, AI

- **MAY / JUN 2024**

ALL QUESTIONS ARE REPEATED !!

- **NOV / DEC 2024**

Q3)

a) **Define Knowledge base and Sentence. Describe in detail about Wumpus World Environment along with brief description to find out the agent. Explain Task Environment. [8]**

Define Knowledge Base and Sentence

- **Knowledge Base (KB):**
  A knowledge base is a collection of sentences or facts expressed in a formal language (like logic) that represents what the agent knows about the world. It stores information that the agent can use to infer new knowledge and make decisions.
- **Sentence:**
  A sentence is a well-formed formula in a logical language representing a statement that can be true or false.

Wumpus World Environment

- The **Wumpus World** is a classic AI environment used to test knowledge-based agents.
- It is a 4x4 grid where an agent explores to find gold while avoiding hazards:
  - **Wumpus:** A monster that kills if the agent enters its cell.
  - **Pits:** Dangerous holes that cause the agent to fall.
- The agent receives percepts:
  - **Stench:** Adjacent to Wumpus.
  - **Breeze:** Adjacent to pits.
  - **Glitter:** In the same cell as gold.
- The agent must infer safe squares using percepts and logical rules.
- The agent's goal is to find gold and exit safely.

Task Environment (PEAS Description)

- **Performance measure:** Maximize gold collected, avoid falling into pits or Wumpus, minimize steps.
- **Environment:** 4x4 grid with pits, Wumpus, gold.
- **Actuators:** Move forward, turn left/right, grab gold, shoot arrow, climb out.
- **Sensors:** Detect breeze, stench, glitter, bump, scream.

b) Represent the followings into First Order Logic form: [10]

i) All employees earning Rs.45000 or more pay tax.

ii) Sita is a marine engineer and she is also an artist.

iii) Children love icecream.

iv) If Humidity is high,temperature is high then a person cannot feel comfortable.

V) Puppies are cute.

If AB and AC are equal, then angle B and C are equal.

ABC is an equilateral triangle.
Represent these facts in predicate logic.

First Order Logic (with quantifiers):

First Order Logic (with quantifiers):

i) **All employees earning Rs.45000 or more pay tax.**

$$\forall x\big(Employee(x) \land Earns(x, \geq 45000) \Rightarrow PaysTax(x)\big)$$

---

ii) **Sita is a marine engineer and she is also an artist.**

(Since it is a statement about a specific individual, no quantifier needed here)

$$MarineEngineer(Sita) \land Artist(Sita)$$

---

iii) **Children love ice cream.**

$$\forall x\big(Child(x) \Rightarrow Loves(x, IceCream)\big)$$

iv) **If humidity is high and temperature is high then a person cannot feel comfortable.**

$$\forall x\big(Person(x) \land HighHumidity \land HighTemperature \Rightarrow \neg Comfortable(x)\big)$$

---

v) **Puppies are cute.**

$$\forall x\big(Puppy(x) \Rightarrow Cute(x)\big)$$

Predicate Logic (facts and rules, typically without explicit quantifiers):

vi) **If AB and AC are equal, then angle B and angle C are equal.**

$$Equal(AB, AC) \Rightarrow Equal(Angle(B), Angle(C))$$

---

vii) **ABC is an equilateral triangle.**

$$EquilateralTriangle(ABC)$$

Summary:

- Sentences (i), (iii), (iv), and (v) are general rules that apply to all relevant objects, so they are written in **First-Order Logic** using universal quantifiers (∀).
- Sentence (ii) is a specific fact about an individual (Sita), so no quantifier is required.
- Sentences (vi) and (vii) represent geometric facts/rules and are expressed in **Predicate Logic** form, often without explicit quantifiers, as they are generally understood to be universally true within the domain.

Q4) a) Write Short notes on followings: [9]

1) **Syntax and Semantics**
2) **Proposition Logic Vs First Order Logic**
3) **Knowledge Engineering Process in First Order Logic**

1) Syntax and Semantics

- **Syntax** refers to the formal structure or rules governing how symbols and formulas are correctly formed in a logic language. It defines the grammar—what strings of symbols count as valid formulas or sentences.
- **Semantics** deals with the meaning or interpretation of these syntactically correct formulas. It assigns truth values to sentences depending on the interpretation or model, defining when a formula is true or false.

Example:

In propositional logic, syntax defines that "P ∧ Q" is valid but " ∧ PQ" is not. Semantics tells us that "P ∧ Q" is true only if both P and Q are true.

2) Propositional Logic Vs First Order Logic (FOL)

Aspect	Propositional Logic	First Order Logic (FOL)
Expressiveness	Deals with simple propositions (true/false)	Deals with objects, predicates, functions, quantifiers
Syntax	Uses propositional variables (P, Q, R)	Uses predicates, variables, functions, quantifiers (∀, ∃)
Semantics	Truth values assigned to propositions	Truth values depend on domain, interpretation of predicates/functions
Quantifiers	None	Universal (∀) and existential (∃) quantifiers used
Example	"P ∧ Q" (It is raining and cold)	"∀x (Bird(x) → Fly(x))" (All birds fly)

3) Knowledge Engineering Process in First Order Logic

Knowledge engineering is the process of creating a knowledge base in FOL. The main steps are:

- **Problem Identification:** Define the domain and scope of knowledge to be represented.
- **Knowledge Acquisition:** Collect facts, rules, and domain knowledge from experts or data.
- **Knowledge Representation:** Express the acquired knowledge formally using FOL syntax with predicates, functions, and quantifiers.
- **Knowledge Validation:** Verify that the represented knowledge is consistent and accurately models the domain.
- **Inference Mechanism:** Implement algorithms to draw conclusions from the knowledge base (e.g., resolution, unification).
- **Maintenance:** Update and refine the knowledge base as new information or changes occur.

b) Show the following Sentences are valid or not. [9]

a) (P upper arrow Q)→(PvQ)

b) (-AB)^(-BvC)→(-AvC)

a) $(P \uparrow Q) \rightarrow (P \vee Q)$

Where $\uparrow$ is the **NAND** (Not AND) operator.

- $P \uparrow Q = \neg(P \wedge Q)$
- The formula says: If **not (P and Q)** then **P or Q**

Check validity with truth values:

P	Q	$P \wedge Q$	$\neg(P \wedge Q) = P \uparrow Q$	$P \vee Q$	$(P \uparrow Q) \rightarrow (P \vee Q)$
T	T	T	F	T	F → T = T
T	F	F	T	T	T → T = T
F	T	F	T	T	T → T = T
F	F	F	T	F	T → F = **F**

- The formula is **false** when P = F and Q = F.
- So, **not valid** (not a tautology).

b) $(\neg AB) \wedge (\neg B \vee C) \rightarrow (\neg A \vee C)$

Rewrite clearly as:

$$(\neg A \wedge B) \wedge (\neg B \vee C) \rightarrow (\neg A \vee C)$$

Check validity:

We need to check if this formula is true under all interpretations.

A	B	C	$\neg A$	$\neg A \wedge B$	$\neg B \vee C$	LHS: $(\neg A \wedge B) \wedge (\neg B \vee C)$	RHS: $\neg A \vee C$	Formula $LHS \rightarrow RH$
T	T	T	F	F	F or T = T	F	F or T = T	T
T	T	F	F	F	F or F = F	F	F or F = F	T
T	F	T	F	F	T or T = T	F	F or T = T	T
T	F	F	F	F	T or F = T	F	F or F = F	T
F	T	T	T	T	F or T = T	T	T or T = T	T
F	T	F	T	T	F or F = F	T ∧ F = F	T or F = T	T
F	F	T	T	F	T or T = T	F	T or T = T	T
F	F	F	T	F	T or F = T	F	T or F = T	T

- **Whenever LHS is false, the implication is true by default.**
- **When LHS is true, RHS is also true.**
- **Hence, the formula is valid.**