Parul® University

ARTIFICIAL INTELLIGENCE II

UNIT-4

Knowledge Representation

Objectives of unit 4

4.1 Knowledge & Knowledge representation issues:

4.1.1 General concepts, Definition and importance of knowledge

4.1.2 Representations and mappings

4.1.3 Approaches to knowledge representation

4.1.4 Frame problem.

4.2 Using predicate knowledge:

4.2.1 Representing instance and Is-a relationship

4.2.2 Computable functions and predicates

4.2.3 Resolutions

4.2.4 Natural deduction.

4.1 Knowledge & Knowledge representation

4.1.1 General concepts, Definition and importance of knowledge

- ***Knowledge representation is a core aspect of Artificial Intelligence*** (AI) and focuses on how knowledge can be structured, stored, and used to enable machines to mimic human decision-making and reasoning.
- **Definition of Knowledge**:
  Knowledge is structured information that is interpreted and contextualized to provide meaning and support decision-making. It encompasses facts, concepts, rules, and relationships about the world.

4.1 Knowledge & Knowledge representation issues:

4.1.1 General concepts, Definition and importance of knowledge

Importance of Knowledge in AI:

- **Foundation of Intelligence**: Knowledge enables reasoning, learning, and problem-solving, which are essential for intelligent behavior.
- **Decision-Making**: AI systems rely on knowledge to make informed decisions, whether in diagnosis, planning, or predictions.
- **Human-AI Interaction**: Representing human-like knowledge allows AI to interact effectively in natural language and understand human contexts.
- **Efficiency**: Properly represented knowledge enables faster computation and more effective problem-solving.

4.1 Knowledge & Knowledge representation issues:

4.1.2 Representations and mappings

- **Representations**: Knowledge representation is the method of encoding information for AI systems to process. It must balance **expressiveness** (capturing complex details) and **computability** (allowing efficient reasoning).
- **Mappings**: Mappings define relationships between:
  - **Real-world entities** and **their representations** in the system (e.g., a concept like "car" mapped to its attributes: speed, color, brand).
  - **Abstract and concrete data** to make reasoning possible (e.g., mapping "health condition" to specific symptoms and treatments).

4.1 Knowledge & Knowledge representation issues:

4.1.2 Representations and mappings…

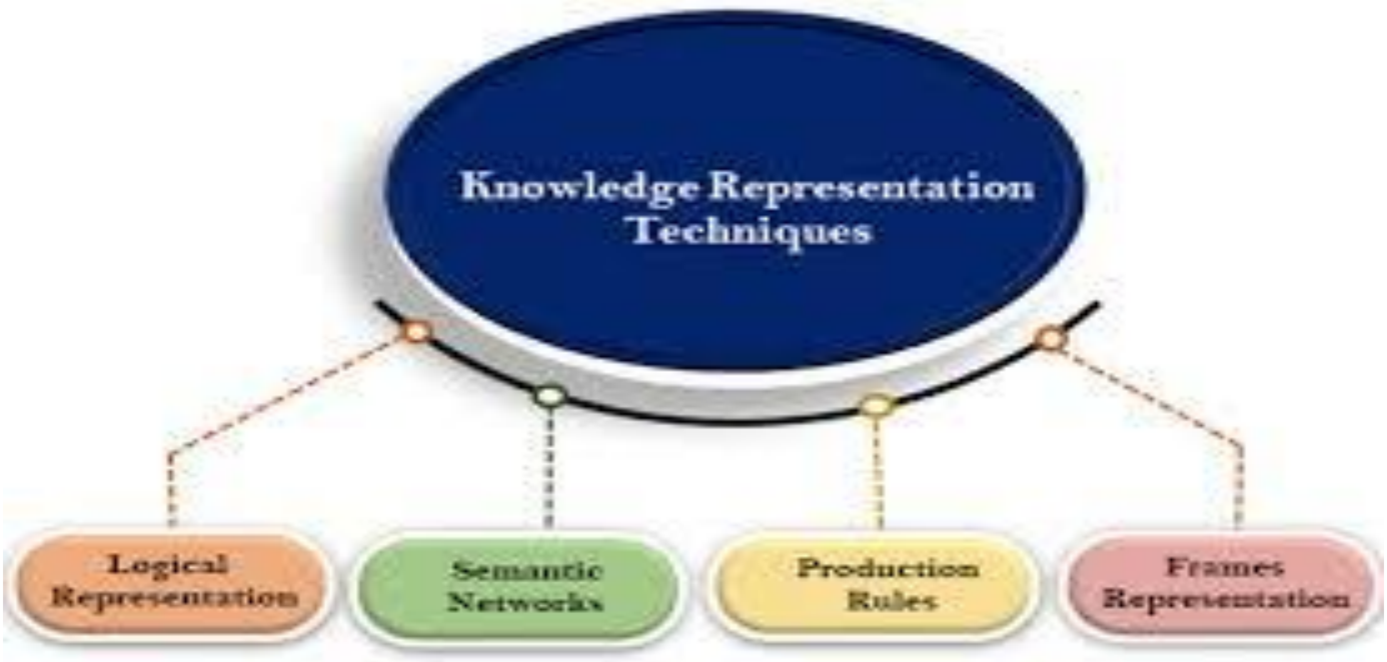
Types of Representations and Mappings:

1. **Symbolic**: Use symbols and rules (e.g., propositional and predicate logic).
2. **Sub-symbolic**: Use neural networks and patterns (e.g., deep learning models).
3. **Hybrid**: Combine symbolic and sub-symbolic approaches.

4.1 Knowledge & Knowledge representation issues:

4.1.3 Approaches to knowledge representation

...



4.1 Knowledge & Knowledge representation issues:

4.1.3 Approaches to knowledge representation …

1. Logical Representation:

- Uses formal logic to represent knowledge.
- Propositions and predicates express facts, while inference rules derive new knowledge.
- Example: Predicate logic like **$\forall x\ (Cat(x) \rightarrow Mammal(x))$**.

Advantages: Precise, unambiguous, and suitable for reasoning.

Challenges: Computationally expensive and may lack the ability to handle uncertainty

4.1 Knowledge & Knowledge representation issues:

4.1.3 Approaches to knowledge representation …

2. Semantic Networks:

- Knowledge is **represented as a graph** where nodes are entities, and edges represent relationships.
- Example: A node "Dog" connected to "Animal" with a "is-a" relationship.

Advantages: Intuitive and visual representation of relationships.

Challenges: Not always efficient for large-scale knowledge bases.

4.1 Knowledge & Knowledge representation issues:

4.1.3 Approaches to knowledge representation …

3. Frame-based Representation:

- Knowledge is **structured as "frames,"** similar to objects in object-oriented programming.
- Each frame has slots (attributes) and fillers (values).
- Example: Frame for "Car" with slots like Make: Toyota, Color: Red.

Advantages: Organizes knowledge hierarchically and allows inheritance. **Challenges**: Can become complex with deeply nested frames.

4.1 Knowledge & Knowledge representation issues:

4.1.3 Approaches to knowledge representation …

4. Production Rules:

- Knowledge is represented as a set of "**if-then**" rules.
- Example: If temperature > 37.5, then Person has fever.

Advantages: Suitable for decision-making and problem-solving.

Challenges: Difficult to maintain when rules grow in number.

4.1 Knowledge & Knowledge representation issues:

4.1.4 Frame Problem

The **frame problem arises in AI** when **trying to represent and update knowledge about a changing world.**

It specifically deals with how to determine which facts remain true after an action is performed.

Representing Change:

1. When an AI performs an action, it needs to know how that action affects the environment.
2. Example: If a robot moves an object from table A to table B, does everything else i the environment remain unchanged?

Handling Non-Changes:

1. Explicitly stating every unaffected aspect after an action is computationally expensive.
2. Example: A robot should infer that moving a box does not affect the color of the room, rather than explicitly updating this fact.

4.1 Knowledge & Knowledge representation issues:

Solutions to the Frame Problem:

1.Successor State Axioms:

1. Encode which properties change and which don't as part of the logic for the action.

2.Default Reasoning:

1. Assume that things remain the same unless explicitly stated otherwise.

3.Qualitative Modeling:

1. Focus on only relevant features and ignore irrelevant details for efficiency.

4.Planning Algorithms:

1. Use specific techniques (like STRIPS in planning) to handle state changes efficiently.

4.2 Using predicate knowledge

Predicate knowledge is a formal representation of facts and rules in AI using predicate logic. Predicate logic (or first-order logic) is more expressive than propositional logic because it allows the use of variables, quantifiers, functions, and relations to describe the world.

Parul® University

DIGITAL LEARNING CONTENT

4.2 Using predicate knowledge

4.2.1 Representing instance and Is-a relationship

- **Instance Relationship (Instance-of):**
  Describes that an object is a specific instance of a class.
  Syntax: Instance(obj, Class)
  Example: Instance(Simba, Lion) (Simba is an instance of the Lion class).
- **Is-a Relationship:** Represents class hierarchy or subclass relationships.
  Syntax: Is-a(Subclass, Superclass)

Example: Is-a(Lion, Mammal) (The Lion class is a subclass of the Mammal class).

4.2 Using predicate knowledge

4.2.1 Representing instance and Is-a relationship

- **Combining Instance and Is-a:** Instance(Simba, Lion) + Is-a(Lion, Mammal) implies Simba is also a Mammal through inheritance.
- **Usage in Reasoning:**

Allows for inheritance of properties: If Is-a(Lion, Mammal) and all mammals have lungs, then Simba, as a lion, has lungs.

4.2 Using predicate knowledge

4.2.2 Computable functions and predicates

- **Functions:**

  A function maps inputs to a specific output. In predicate logic, functions are used to compute relationships or values.

  Syntax: f(x1, x2, ..., xn)

  Example: Father(John) could return Mark (Mark is John's father).

  Age(Simba) could return 5.

- **Predicates:**

  Predicates evaluate whether a certain property or relationship holds true.

  Syntax: P(x1, x2, ..., xn)

  Example: Larger(Elephant, Dog) evaluates whether an elephant is larger than a dog (returns True or False).

  HasParent(Simba, Mufasa) checks if Simba has Mufasa as a parent.

4.2 Using predicate knowledge

4.2.2 Computable functions and predicates

- **Computable Functions:**

  Some predicates and functions can be made computable by **defining algorithms for their evaluation**.

  Example: Distance(Point1, Point2) calculates the Euclidean distance.

  IsEven(Number) computes whether a number is even.

4.2 Using predicate knowledge

4.2.3 Resolutions

Resolution is a rule of inference used in predicate logic to derive conclusions by refuting contradictions. It's particularly powerful in automated theorem proving.**[A rule of inference that creates new clauses from existing clauses]**

How It Works:

1. Convert all statements into **clausal form** (a disjunction of literals).
2. Negate the statement to be proved and add it to the knowledge base.
3. Use **unification** to resolve clauses and derive new conclusions.
4. If a contradiction is found, the original statement is true.

4.2 Using predicate knowledge

4.2.3 Resolutions

Example:

Knowledge Base: ∀x (Cat(x) → Mammal(x)) (All cats are mammals).
Cat(Tom) (Tom is a cat).

Convert to Clausal Form: ¬Cat(x) ⋁ Mammal(x)
Cat(Tom)

Negate the Query: Suppose we want to prove Mammal(Tom).
Negate it: ¬Mammal(Tom).

Resolve: ¬Cat(x) ⋁ Mammal(x) + Cat(Tom) → Mammal(Tom).

Applications:

Used in Prolog and other logic programming languages.
Essential for logical inference engines in AI.

4.2 Using predicate knowledge

4.2.4 Natural Deduction

Natural deduction is a method of reasoning **where conclusions are derived directly from premises using inference rules**. It closely resembles human logical reasoning.

Core Inference Rules:

Modus Ponens: If P → Q (if P implies Q) and P is true, then Q is true.

Example:

Premises: If it rains, the ground will be wet (Rain → Wet) and It rains (Rain).

Conclusion: The ground is wet (Wet).

4.2 Using predicate knowledge

4.2.4 Natural Deduction

Modus Tollens: If P → Q and ¬Q, then ¬P.

Example:

Premises: If it rains, the ground will be wet (Rain → Wet) and The ground is not wet (¬Wet).

Conclusion: It does not rain (¬Rain).

Universal Instantiation: From ∀x P(x) (for all x, P holds), infer P(a) for a specific a.

Example:

Premise: All humans are mortal (∀x Human(x) → Mortal(x)).

Conclusion: Socrates is mortal (Mortal(Socrates)) since Human(Socrates).

4.2 Using predicate knowledge

4.2.4 Natural Deduction

Existential Generalization: From P(a), infer ∃x P(x) (there exists an x for which P holds).

Example:

Premise: Mortal(Socrates).

Conclusion: ∃x Mortal(x) (There exists someone who is mortal).

Advantages:

Natural deduction is intuitive and mirrors how humans reason.

It allows for step-by-step reasoning to explain AI decisions.

Applications in AI:

Used in expert systems and rule-based systems for deriving conclusions.

Helps in building interpretable AI systems.

Unit 4

Conclusion

Predicate logic provides a robust framework for representing and reasoning with knowledge in AI. It captures relationships, enables reasoning through resolution and natural deduction, and handles comple structures like is-a hierarchies and computable functions. These techniques are foundational in knowledge-based AI systems, theorem proving, and logic programming languages like Prolog.

Unit 4

Multiple Choice Questions (MCQs)

1. What is Knowledge Representation in AI?
a) Storing raw data in a database
b) Representing information about the world in a form that a computer can utilize
c) Programming a machine to perform mathematical calculations
d) None of the above
Answer: b) Representing information about the world in a form that a computer can utilize

2. Which of the following is NOT an approach to knowledge representation?
a) Logical representation
b) Procedural representation
c) Heuristic representation
d) Statistical representation
Answer: d) Statistical representation

3. What is the frame problem in knowledge representation?
a) Difficulty in storing large amounts of data
b) The problem of representing and updating changing knowledge
c) The issue of machine learning efficiency
d) A challenge in database indexing
Answer: b) The problem of representing and updating changing knowledge

4. The Is-a relationship is mainly used to represent:
a) Instance relationships
b) Hierarchical relationships
c) Functional dependencies
d) Statistical dependencies
Answer: b) Hierarchical relationships

5. What does "Resolution" refer to in predicate logic?
a) A technique for simplifying problems
b) A rule of inference used for proving logical conclusions
c) A database management concept
d) A method of programming languages
Answer: b) A rule of inference used for proving logical conclusions

6. Which of the following is NOT a key property of a good knowledge representation system?
a) Completeness
b) Redundancy
c) Computational efficiency
d) Expressiveness
Answer: b) Redundancy

7. Which logic is primarily used in knowledge representation?
a) Propositional logic

b) Predicate logic
c) Fuzzy logic
d) All of the above
Answer: d) All of the above

8. What is the main purpose of Natural Deduction in logic?
a) To prove theorems systematically
b) To store knowledge in structured formats
c) To represent uncertain knowledge
d) To classify objects into categories
Answer: a) To prove theorems systematically

9. What is a computable function in knowledge representation?
a) A function that can be computed using an algorithm
b) A function that represents physical systems
c) A function that cannot be computed
d) A function that depends on human input
Answer: a) A function that can be computed using an algorithm

10. Which representation is most commonly used for machine reasoning?
a) Semantic networks
b) Predicate logic
c) Frames
d) Scripts
Answer: b) Predicate logic

Definition

Knowledge Representation:
It is the process of encoding knowledge about the world into a format that a machine can process and reason about. It includes various techniques such as predicate logic, semantic networks, frames, and rules to store, retrieve, and manipulate knowledge efficiently.

Frame Problem:
The frame problem refers to the challenge of representing and updating knowledge about a dynamic world, specifically determining which facts remain unchanged when an action occurs.

Predicate Logic:
Predicate logic is a formal system in which statements are expressed using predicates, quantifiers, and variables to represent relationships and properties of objects in a domain.

Natural Deduction:
Natural deduction is a method of formal reasoning that derives conclusions from premises using inference rules, closely mimicking human logical reasoning.

▯ **Semantic Network:**
A semantic network is a graphical representation of knowledge that uses nodes to represent concepts and edges to depict relationships between those concepts.

10-Mark Question with Answer

Q: Explain different approaches to knowledge representation with examples.

Answer:

Knowledge representation in Artificial Intelligence (AI) is crucial for enabling machines to store, retrieve, and reason about information. The main approaches are:

1. **Logical Representation:**
   - Uses propositional and predicate logic to represent knowledge formally.
   - Example: "All humans are mortal" can be represented as:
     $\forall x\ (Human(x) \rightarrow Mortal(x))$
2. **Semantic Networks:**
   - Represents knowledge using nodes (concepts) and edges (relationships).
   - Example: A "Dog" is an "Animal," and a "Cat" is also an "Animal."
3. **Frames:**
   - Uses a structured data representation with slots and values.
   - Example: A "Car" frame may include attributes like "Color," "Model," "Year."
4. **Production Rules:**
   - Represents knowledge as IF-THEN rules.
   - Example: IF it is raining, THEN take an umbrella.
5. **Ontology-based Representation:**
   - Defines a hierarchical structure of concepts with well-defined relationships.
   - Example: "Mammals" are a subset of "Animals" with properties like "Warm-blooded."

Each approach has its own advantages and limitations, and the choice depends on the specific problem being solved.

Q: Explain the role of predicate logic in knowledge representation with examples.

Answer:

Predicate logic is a powerful tool used in knowledge representation to express complex relationships and rules about the world. It extends propositional logic by introducing quantifiers and predicates, making it more expressive and useful in AI.

Components of Predicate Logic:

1. **Constants** – Represent objects (e.g., "John," "Apple").
2. **Variables** – Represent general entities (e.g., x, y).
3. **Predicates** – Describe properties or relationships (e.g., Loves(John, Mary)).
4. **Quantifiers** – Universal (∀) and existential (∃) quantifiers.

Examples of Predicate Logic in AI:

1. **Fact Representation:**
   - "All humans are mortal" → ∀x (Human(x) → Mortal(x))
2. **Instance Relationship:**
   - "Socrates is a human" → Human(Socrates)
   - Using Modus Ponens:
     - ∀x (Human(x) → Mortal(x))
     - Human(Socrates)
     - ∴ Mortal(Socrates)
3. **Expressing Relationships:**
   - "John loves Mary" → Loves(John, Mary)
   - "Mary loves Pizza" → Loves(Mary, Pizza)
4. **Rule-based Reasoning:**
   - "If it is raining, then the ground is wet" → ∀x (Raining(x) → Wet(Ground))

Advantages of Predicate Logic in AI:

- **Expressiveness** – Can represent complex relationships.
- **Logical Inference** – Enables reasoning using rules.
- **Universality** – Applicable across various AI domains, including expert systems and natural language processing.