Unit 8: Knowledge Representation and Propositional Logic

1. Knowledge-Based Agents

1.1 What is a Knowledge-Based Agent?

A knowledge-based agent is an intelligent system that uses knowledge about its environment to make decisions and take actions. It operates by reasoning with stored knowledge rather than just following pre-programmed responses.

1.2 Main Components of a Knowledge-Based Agent

A. Knowledge Base (KB)

- The central repository that stores facts, rules, and information about the world
- Contains domain-specific knowledge in a structured format
- Can be updated with new information over time
- Example: A medical diagnosis system's KB contains symptoms, diseases, and their relationships

B. Inference Engine

- The reasoning mechanism that derives new knowledge from existing knowledge
- Applies logical rules to draw conclusions
- Makes decisions based on the KB

1.3 Steps Performed by a Knowledge-Based Agent

1. **TELL**: Add new information to the Knowledge Base
    - Agent perceives the environment
    - Converts perceptions into knowledge representation
    - Stores this knowledge in the KB
2. **ASK**: Query the Knowledge Base for information
    - Agent needs to make a decision
    - Queries the KB using the inference engine
    - Retrieves relevant information
3. **INFER**: Use reasoning to derive new knowledge
    - Apply logical rules to existing knowledge
    - Generate new conclusions
4. **ACT**: Take action based on the derived knowledge
    - Select the best action based on inferred information
    - Execute the action in the environment

Example:

Environment: Robot in a room with objects
1. TELL: "There is a ball in front of me"
2. ASK: "Is there an obstacle ahead?"
3. INFER: "Ball is an obstacle, so yes"
4. ACT: "Turn left to avoid the ball"

1.4 Difference Between Problem-Solving Agent and Knowledge-Based Agent

Aspect	Problem-Solving Agent	Knowledge-Based Agent
Approach	Uses search algorithms and heuristics	Uses logical reasoning and inference
Representation	States and actions	Knowledge in logical form
Decision Making	Based on state space exploration	Based on logical inference
Flexibility	Limited to specific problem types	Can handle diverse scenarios
Learning	Typically doesn't learn	Can update KB with new knowledge
Example	Pathfinding robot using A* algorithm	Medical diagnosis expert system
Memory	Remembers only current path	Maintains comprehensive KB

Practical Example:

- **Problem-Solving Agent**: GPS navigation finding shortest route (pure algorithmic approach)
- **Knowledge-Based Agent**: AI assistant answering "Should I carry an umbrella?" by reasoning about weather, location, and time

---

2. Knowledge Representation Language

2.1 Logic-Based Representation

Logic-based representation uses formal logic to represent knowledge in a precise, unambiguous manner.

Advantages:

- Precise and unambiguous
- Supports reasoning and inference
- Well-defined semantics
- Can verify correctness mathematically

Types:

- Propositional Logic (covered in this unit)
- First-Order Logic
- Temporal Logic
- Modal Logic

---

3. Propositional Logic

3.1 What is Propositional Logic?

Propositional logic is a branch of logic that deals with propositions (statements) that can be either TRUE or FALSE, and the logical relationships between them.

Key Features:

- Simplest form of logic
- Deals with declarative sentences
- Uses logical connectives to build complex statements
- Foundation for more advanced logic systems

3.2 Logic as Expressions

In propositional logic, we represent statements as symbols and combine them using logical operators.

Basic Elements:

- **Atomic Propositions**: Simple statements represented by symbols (P, Q, R, etc.)
  - P: "It is raining"
  - Q: "The ground is wet"
- **Compound Propositions**: Complex statements formed by combining atomic propositions
  - $P \land Q$: "It is raining AND the ground is wet"

3.3 Atomic and Compound Propositions

Atomic Propositions

Simple, indivisible statements that cannot be broken down further.

Examples:

- P: "The sun is shining"
- Q: "Birds are singing"
- R: "Temperature is 25°C"

Compound Propositions

Statements formed by combining atomic propositions using logical connectives.

Examples:

- $P \land Q$: "The sun is shining AND birds are singing"
- $P \lor R$: "The sun is shining OR temperature is 25°C"
- $\neg P$: "The sun is NOT shining"

---

4. Logical Connectives and Operators

4.1 Negation (NOT) - Symbol: ¬ or ~

Reverses the truth value of a proposition.

Truth Table:

P	¬P
T	F
F	T

Example:

- P: "It is day"
- ¬P: "It is NOT day" (meaning it's night)

4.2 Conjunction (AND) - Symbol: ∧

True only when BOTH propositions are true.

Truth Table:

P	Q	P ∧ Q
T	T	T
T	F	F
F	T	F
F	F	F

Example:

- P: "I have money"
- Q: "The shop is open"
- P ∧ Q: "I have money AND the shop is open" (I can buy something)

4.3 Disjunction (OR) - Symbol: ∨

True when AT LEAST ONE proposition is true.

Truth Table:

P	Q	P ∨ Q
T	T	T
T	F	T
F	T	T
F	F	F

Example:

- P: "I study hard"
- Q: "I am naturally talented"
- P ∨ Q: "I study hard OR I am naturally talented" (I will pass)

4.4 Implication (IF-THEN) - Symbol: → or ⇒

Represents "if P then Q". False only when P is true and Q is false.

Truth Table:

P	Q	P → Q
T	T	T
T	F	F
F	T	T
F	F	T

Example:

- P: "It rains"
- Q: "The ground gets wet"
- P → Q: "IF it rains THEN the ground gets wet"

Important Note: When P is false, the implication is always true (vacuously true).

4.5 Biconditional (IF AND ONLY IF) - Symbol: ↔ or ⇔

True when both propositions have the same truth value.

Truth Table:

P	Q	P ↔ Q
T	T	T
T	F	F
F	T	F
F	F	T

Example:

- P: "Triangle has 3 equal sides"
- Q: "Triangle is equilateral"
- P ↔ Q: "A triangle has 3 equal sides IF AND ONLY IF it is equilateral"

---

5. Truth Tables

5.1 What is a Truth Table?

A truth table is a mathematical table that shows all possible truth values for a logical expression.

Uses:

- Verify logical equivalences
- Prove tautologies and contradictions
- Analyze complex logical expressions
- Determine validity of arguments

5.2 Creating Truth Tables

Steps:

1. List all variables
2. Create rows for all possible combinations (2^n rows for n variables)
3. Evaluate sub-expressions
4. Compute final result

Example: (P ∧ Q) → R

P	Q	R	P ∧ Q	(P ∧ Q) → R
T	T	T	T	T
T	T	F	T	F
T	F	T	F	T
T	F	F	F	T
F	T	T	F	T
F	T	F	F	T
F	F	T	F	T
F	F	F	F	T

---

6. Important Logical Concepts

6.1 Tautology

A proposition that is ALWAYS TRUE, regardless of the truth values of its variables.

Example:

- P ∨ ¬P (Either it's raining or it's not raining - always true!)

Truth Table:

P	¬P	P ∨ ¬P
T	F	T
F	T	T

More Examples:

- P → P: "If P then P" (always true)
- (P ∧ Q) → P: "If P and Q, then P" (always true)

6.2 Contradiction

A proposition that is ALWAYS FALSE.

Example:

- P ∧ ¬P (It's raining and not raining simultaneously - impossible!)

Truth Table:

P	¬P	P ∧ ¬P
T	F	F
F	T	F

6.3 Syllogism

A form of logical reasoning where a conclusion is drawn from two premises.

Classic Example:

```
Premise 1: All humans are mortal (If human → mortal)
Premise 2: Socrates is human
Conclusion: Therefore, Socrates is mortal
```

In Propositional Logic:

```
P → Q  (If it rains, the ground is wet)
P      (It is raining)
------
∴ Q    (Therefore, the ground is wet)
```

This is called **Modus Ponens** (a valid inference rule).

---

7. Logical Equivalence Laws

7.1 Commutativity

The order of operands doesn't matter.

Conjunction:

- $P \land Q \equiv Q \land P$
- Example: "I am hungry AND tired" = "I am tired AND hungry"

Disjunction:

- $P \lor Q \equiv Q \lor P$
- Example: "Coffee OR tea" = "Tea OR coffee"

7.2 Associativity

Grouping of operands doesn't matter.

Conjunction:

- $(P \land Q) \land R \equiv P \land (Q \land R)$
- Example: "(Red AND blue) AND green" = "Red AND (blue AND green)"

Disjunction:

- $(P \lor Q) \lor R \equiv P \lor (Q \lor R)$
- Example: "(Monday OR Tuesday) OR Wednesday" = "Monday OR (Tuesday OR Wednesday)"

7.3 Distributivity

One operator distributes over another.

AND distributes over OR:

- $P \land (Q \lor R) \equiv (P \land Q) \lor (P \land R)$
- Example: "I'll eat pizza AND (salad OR soup)" = "(I'll eat pizza AND salad) OR (I'll eat pizza AND soup)"

OR distributes over AND:

- $P \lor (Q \land R) \equiv (P \lor Q) \land (P \lor R)$

7.4 De Morgan's Theorem

One of the most important laws for simplifying logical expressions.

First Law:

- $\neg(P \land Q) \equiv \neg P \lor \neg Q$
- "NOT (P AND Q)" = "NOT P OR NOT Q"

Example:

- $\neg(\text{Sunny} \land \text{Warm}) = \neg\text{Sunny} \lor \neg\text{Warm}$
- "It's not (sunny and warm)" = "It's not sunny or not warm"
- "It's not a nice day" = "It's either cloudy or cold (or both)"

Second Law:

- $\neg(P \lor Q) \equiv \neg P \land \neg Q$
- "NOT (P OR Q)" = "NOT P AND NOT Q"

Example:

- $\neg(\text{Coffee} \lor \text{Tea}) = \neg\text{Coffee} \land \neg\text{Tea}$
- "I don't want (coffee or tea)" = "I don't want coffee and I don't want tea"
- "I want neither"

Proof by Truth Table:

P	Q	P ∧ Q	¬(P ∧ Q)	¬P	¬Q	¬P ∨ ¬Q
T	T	T	F	F	F	F
T	F	F	T	F	T	T
F	T	F	T	T	F	T
F	F	F	T	T	T	T

Columns 4 and 7 are identical, proving ¬(P ∧ Q) ≡ ¬P ∨ ¬Q

---

8. Practical Examples of Logical Reasoning

Example 1: Daily Life Scenario

Statements:

- P: "I wake up early"
- Q: "I catch the bus"
- R: "I reach office on time"

Given Knowledge:

1. P → Q (If I wake up early, I catch the bus)
2. Q → R (If I catch the bus, I reach office on time)
3. P (I woke up early today)

Inference:

- From 3 and 1: Q is true (I caught the bus)
- From Q and 2: R is true (I reached office on time)
- **Conclusion**: I reached office on time ✓

Example 2: Academic Scenario

Statements:

- P: "Student studies regularly"
- Q: "Student completes assignments"
- R: "Student passes the exam"

Given:

- (P ∧ Q) → R
- P is true
- Q is true

Inference:

- P ∧ Q is true
- Therefore, R is true (Student will pass)

Example 3: Shopping Decision

Statements:

- P: "Product is good quality"
- Q: "Product is affordable"
- R: "I will buy the product"

Rule: $(P \land Q) \rightarrow R$

Scenario 1:

- P = True (Good quality)
- Q = False (Not affordable)
- $P \land Q$ = False
- Conclusion: I won't buy (R can be true or false, but likely false)

Scenario 2:

- P = True (Good quality)
- Q = True (Affordable)
- $P \land Q$ = True
- Conclusion: I will buy (R = True)

Example 4: Medical Diagnosis

Statements:

- P: "Patient has fever"
- Q: "Patient has cough"
- R: "Patient has flu"

Knowledge Base:

- $(P \land Q) \rightarrow R$ (Fever and cough indicate flu)
- P is observed (Patient has fever)
- Q is observed (Patient has cough)

Inference:

- $P \land Q$ = True
- Therefore, R = True (Patient likely has flu)

---

9. Inference in Propositional Logic

9.1 What is Inference?

Inference is the process of deriving new conclusions from existing knowledge using logical rules.

9.2 Common Inference Rules

A. Modus Ponens (Law of Detachment)

```
P → Q
P
------
∴ Q
```

Example:

- If it rains (P), the ground is wet (Q)
- It is raining (P)
- **Therefore**, the ground is wet (Q)

B. Modus Tollens

```
P → Q
¬Q
------
∴ ¬P
```

Example:

- If I study (P), I will pass (Q)
- I did not pass (¬Q)
- **Therefore**, I did not study (¬P)

C. Hypothetical Syllogism (Chain Rule)

```
P → Q
Q → R
------
∴ P → R
```

Example:

- If I save money (P), I can buy a bike (Q)
- If I buy a bike (Q), I can travel easily (R)
- **Therefore**, if I save money (P), I can travel easily (R)

D. Disjunctive Syllogism

```
P ∨ Q
¬P
------
∴ Q
```

Example:

- The problem is hardware (P) or software (Q)
- It's not hardware (¬P)
- **Therefore**, it's software (Q)

9.3 Inference Example: Crime Investigation

Knowledge Base:

1. (Fingerprints ∧ Motive) → Guilty
2. Fingerprints (found at scene)
3. Motive (had reason)
4. Guilty → Arrest

Inference Chain:

- From 2 and 3: Fingerprints ∧ Motive = True
- From 1: Guilty = True
- From 4: Arrest = True
- **Conclusion**: Person should be arrested

---

10. Answer Extraction System

10.1 What is an Answer Extraction System?

A system that uses logical inference to extract specific answers from a knowledge base in response to queries.

10.2 How It Works

1. **Input**: User query in natural language
2. **Conversion**: Transform query into logical form
3. **Search**: Match query against knowledge base
4. **Inference**: Apply logical rules to derive answer
5. **Output**: Return answer in natural language

10.3 Simple Example: University Information System

Knowledge Base:

```
KB1: Student(John) → Person(John)
KB2: Person(John)
KB3: Course(Math) ∧ Enrolled(John, Math)
KB4: Enrolled(X, Y) → Studies(X, Y)
KB5: Grade(John, Math, A)
```

Query 1: "Is John a student?"

- Match: Student(John)
- From KB1 and KB2: Can infer Student(John) is related
- **Answer**: Yes

Query 2: "What does John study?"

- Match: Enrolled(John, Math) from KB3
- From KB4: Studies(John, Math)
- **Answer**: Math

Query 3: "What grade did John get in Math?"

- Direct match: KB5
- **Answer**: A

10.4 Practical Example: Restaurant Recommendation System

Knowledge Base:

```
1. Italian(Restaurant_A) ∧ Expensive(Restaurant_A)
2. Chinese(Restaurant_B) ∧ Cheap(Restaurant_B)
3. Italian(Restaurant_C) ∧ Cheap(Restaurant_C)
4. (Budget_Low ∧ Likes_Italian) → Recommend(Restaurant_C)
5. (Budget_Low ∧ Likes_Chinese) → Recommend(Restaurant_B)
```

User Query: "I like Italian food and have a low budget"

- Convert to logic: Budget_Low ∧ Likes_Italian
- Match with KB4: Recommend(Restaurant_C)
- **Answer**: "I recommend Restaurant C - it serves Italian food and is affordable"

10.5 Medical Expert System Example

Knowledge Base:

1. (Fever ∧ Cough ∧ Fatigue) → Flu
2. (Fever ∧ Rash) → Measles
3. (Headache ∧ Stiff_Neck) → Meningitis
4. Flu → Medicine(Paracetamol)
5. Measles → Action(Quarantine)

User Input: "Patient has fever, cough, and fatigue"

- Convert: Fever ∧ Cough ∧ Fatigue = True
- From KB1: Flu = True
- From KB4: Medicine(Paracetamol)
- **Answer**: "Patient likely has flu. Recommend Paracetamol."

---

11. Key Takeaways

Knowledge-Based Agents

- Use stored knowledge to make intelligent decisions
- Components: Knowledge Base and Inference Engine
- More flexible than simple problem-solving agents

Propositional Logic

- Foundation of logical reasoning
- Uses propositions (statements) that are true or false
- Combines propositions using logical connectives

Logical Operations

- **NOT (¬)**: Reverses truth value
- **AND (∧)**: True when both are true
- **OR (∨)**: True when at least one is true
- **Implication (→)**: If-then relationship
- **Biconditional (↔)**: If and only if

Important Laws

- **Commutativity**: Order doesn't matter
- **Associativity**: Grouping doesn't matter
- **Distributivity**: One operation distributes over another
- **De Morgan's**: Rules for negating AND/OR expressions

Practical Applications

- Expert systems (medical, legal)
- Decision support systems

- Automated reasoning
- Question-answering systems
- Natural language processing

---

12. Practice Problems

Problem 1

Given: $P \rightarrow Q$, $Q \rightarrow R$, P is true **Question**: Is R true? **Answer**: Yes (by chain rule: $P \rightarrow R$)

Problem 2

Simplify: $\neg(P \lor Q)$ **Answer**: $\neg P \land \neg Q$ (De Morgan's Law)

Problem 3

Create truth table for: $(P \land Q) \rightarrow R$

P	Q	R	P ∧ Q	(P ∧ Q) → R
T	T	T	T	T
T	T	F	T	F
T	F	T	F	T
T	F	F	F	T
F	T	T	F	T
F	T	F	F	T
F	F	T	F	T
F	F	F	F	T

Problem 4

Scenario: "If it's a weekend (W) and the weather is good (G), I'll go hiking (H)" Write in logical form: $(W \land G) \rightarrow H$

Given: It's Saturday (W = True), Weather is sunny (G = True) **Question**: Will I go hiking? **Answer**: Yes, H = True

---

Summary

This unit covers the fundamentals of representing and reasoning with knowledge using propositional logic. Knowledge-based agents use logical inference to make intelligent decisions based on stored facts and rules. Propositional logic provides a formal framework for representing statements and their relationships, enabling systematic reasoning and automated decision-making in AI systems.

The concepts learned here are foundational for:

- Artificial Intelligence
- Expert Systems
- Automated Reasoning
- Natural Language Processing
- Decision Support Systems