Unit 3
Knowledge Representation and Reasoning
(10 Hours)

https://github.com/brylevkirill/notes/tree/master

3.1 Definition and importance of Knowledge, Issues in Knowledge Representation

3.2. Knowledge Representation Systems: Semantic Nets, Frames, Conceptual Dependencies, Scripts, Rule Based Systems (Production System), Propositional Logic, Predicate Logic

3.3.Propositional Logic(PL): Syntax, Semantics, Formal logic connectives, truth tables, tautology, validity, well-formed formula, Inference using Resolution,

3.4 Backward Chaining and Forward Chaining

3.5 Predicate Logic: FOPL, Syntax, Semantics, Quantification, Inference with FOPL, Inference using resolution

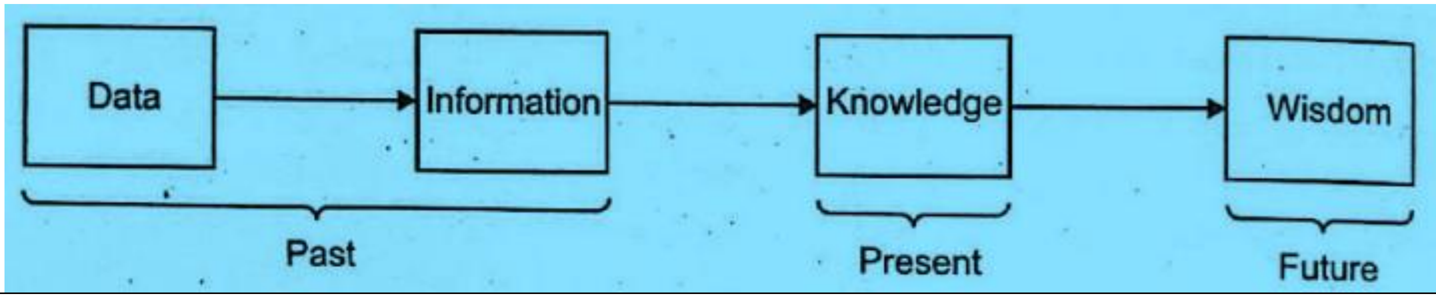
3.6 Bayes' Rule and its use, Bayesian Networks

3.7 Fuzzy Logic

Sanjeev Thapa, Er. DevOps, SRE, CKA, RHCSA, RHCE, RHCSA-Openstack, MTCNA, MTCTCE, UBSRS, HEv6, Research Evangelist

4.0 Introduction

Knowledge is a progression that starts with data which is of limited utility. Data when processed becomes information, information when interpreted or evaluated becomes knowledge and understanding, of the principles embodied with the knowledge is wisdom.



Data → Information → Knowledge → Wisdom (DIKW Model)

Imagine a health monitoring system used in a smart hospital.

🧩 **Step 1 — Data (Past)**
Raw, unprocessed facts with *no meaning* by themselves.

Example (Data):

- Temperature readings: **101.4°F, 99.8°F, 102.1°F**
- Heart rate values: **112 bpm, 108 bpm, 118 bpm**
- Oxygen level readings: **94%, 92%, 95%**

These are **just numbers** collected by sensors.

➡ They do *not* tell us what is happening.

🧩 **Step 2 — Information (Past)**
Data processed to give **context and meaning**.

Example (Information):

- "Patient's temperature is **consistently above 101°F**."
- "Heart rate is **higher than normal**."
- "Oxygen level is **slightly low**."

Now the system interprets data → useful patterns.

➡ Still no decision, but the system *understands what the data represents*.

🧩 **Step 3 — Knowledge (Present)**
Using **rules, models, and experience** to interpret information.

Example (Knowledge):
The AI system applies medical rules:

- IF fever > 101°F AND heart rate > 110 THEN → fever with tachycardia
- IF oxygen < 94% THEN → possible respiratory problem
- IF fever + cough + low oxygen THEN → **suspected pneumonia**

Here AI combines information + stored rules.

➡ AI "knows" how symptoms relate to illnesses.

🧩 **Step 4 — Wisdom (Future)**
Ability to make **sound decisions or recommendations**.

Example (Wisdom):
Based on knowledge, AI suggests:

- "Patient may require a chest X-ray."
- "Start oxygen support immediately."
- "Alert the doctor with high priority."
- "Predict 24-hour risk of deterioration: HIGH."

☆ **In Contrast to AI**

The DIKW model naturally fits with **how AI systems operate**:

DIKW Level	**AI Perspective**
Data	Raw sensor input (temperature, images, GPS points)
Information	Feature extraction, preprocessing (e.g., detect fever, analyze pixels)
Knowledge	Inference rules, trained models, patterns learned
Wisdom	AI decision-making, predictions, recommendations

AI transforms data → wisdom using:

- Logic

Sanjeev Thapa, Er. DevOps, SRE, CKA, RHCSA, RHCE, RHCSA-Openstack, MTCNA, MTCTCE, UBSRS, HEv6, Research Evangelist

Wisdom = **Choosing the best action** based on knowledge. ➡ This is closest to intelligent decision-making.	• Rules • Machine learning • Neural networks • Inference engines

☆ 📌 Example 2: Smart Agriculture Monitoring System (Farmer's AI Tool)

System: AI system monitoring crops in Rupandehi or Chitwan.

1 Data (Past)

Collected by sensors and drones:

- Soil moisture readings
- Temperature values
- Leaf color pixels from images
- Rainfall logs

These are *raw, unprocessed*.

2 Information (Past)

Processed:

- Moisture is **below safe level**
- Crop leaf color = **yellow patches**
- Temperature = **higher than normal**
- Rainfall = **insufficient**

The system now understands crop conditions.

3 Knowledge (Present)

AI applies agriculture rules and ML models:

- "Low moisture + high temperature = risk of dehydration."
- "Yellow patches = early nitrogen deficiency."
- "Low rainfall = irrigation needed."

AI forms meaningful **agricultural knowledge**.

4 Wisdom (Future)

System gives **actionable advice**:

- "Irrigate the field within 4 hours."
- "Add nitrogen-rich fertilizer tomorrow morning."
- "Schedule shade nets during peak heat."
- "Predict yield loss: 15% if untreated."

This is **wisdom**—the ability to decide **what to do next**.

⭐ Short Summary for Both Examples

DIKW Stage	Traffic System	Agriculture System
Data	camera counts, GPS logs	soil moisture, leaf pixels
Information	jam detected, slow speed	crop stress found
Knowledge	causes jam, predicts patterns	nutrient deficiency rules
Wisdom	change signals, reroute traffic	irrigate, fertilize, protect crop

✅ What is Knowledge Representation (KR)?

Knowledge Representation is a field of Artificial Intelligence concerned with:

- **How knowledge about the world is structured**
- **How it is stored in a format that computers can use**
- **How it can be used for reasoning and decision making**

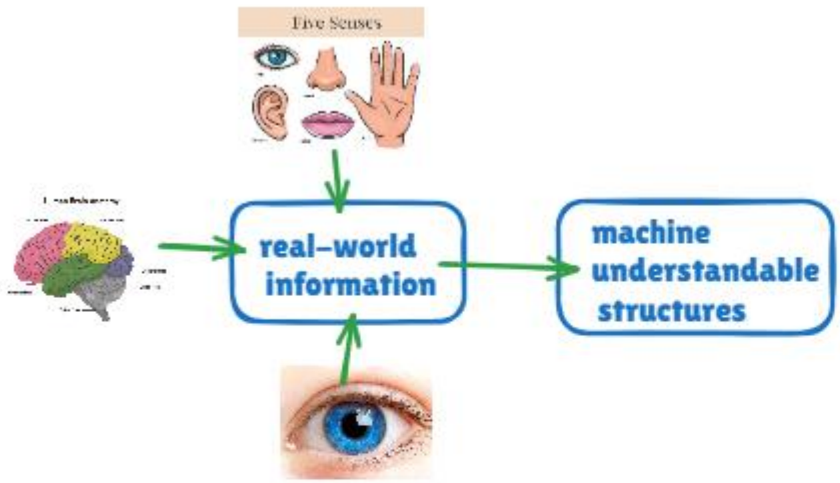
In simple terms:

"KR is the way an intelligent system understands, organizes, and uses knowledge to solve problems."

A KR system should allow a machine to behave **intelligently**, similar to a human who uses facts, rules, and experience to make decisions.

✅ Why Knowledge Representation is Needed?

Computers do not understand the real world naturally.
AI systems must convert ***real-world information → machine-understandable structures***.



Knowledge Representation helps AI systems to:

- Understand problems
- Reason logically
- Draw conclusions
- Take actions
- Learn from previous knowledge

⭐ Key Goals of KR

1. **Represent the real world**
   – Objects, people, events, locations, properties, relationships
2. **Enable intelligent reasoning**
   – Derive new facts from known facts
3. **Support problem solving**
   – Diagnostics, planning, decision making
4. **Efficient storage & retrieval**
   – Organized knowledge for fast reasoning
5. **Provide a basis for communication**
   – Between AI systems and humans

⭐ Three Main Components in KR

1. Facts

Truths about the world.
Example:
“Patient John has fever.”

2. Instances

Specific objects or examples.
Example:
John, **Laptop-01**, **Room-215**

3. Classes (Concepts)

Groups of similar objects.
Example:

- **Human**
- **Disease**
- **Electronic device**

⭐ Suitable Real-World Example (Hospital KR Example)

Suppose we represent knowledge for a **hospital diagnosis system**.

Facts:

- Patient(Ashok) has Fever
- Patient(Ashok) has Cough
- Temperature(Ashok) = 102°F

Rules:

IF Fever AND Cough → Possible Influenza
IF Temperature > 101°F → High Fever

Instance:

- Ashok is an instance of the class *Patient*.

Class hierarchy:

- Patient → Human
- Influenza → Viral Disease
- Disease → Medical Condition

Reasoning:

From the rules, the system can infer:

“Ashok may have influenza.”

This is the purpose of KR:
Represent → Reason → Conclude.

⭐ Relationship Between KR and Reasoning

Knowledge Representation is incomplete without **reasoning mechanisms**.

Reasoning allows AI to:

- Deduce new facts
- Check consistency
- Solve problems
- Diagnose conditions
- Make predictions

Example:

If the system knows:

- "All humans are mortal"
- "Sita is human"

It can reason:
→ "Sita is mortal."

⭐ Requirements of Good Knowledge Representation

✔ 1. Expressive

Meaning:
The representation must describe many types of knowledge such as:

- Objects
- Relations
- Events
- Rules
- Time
- Causes

Example:
A medical AI must represent:

- Objects → *Patient, Doctor, Virus, Medicine*
- Relations → *hasSymptom(Patient, Fever)*
- Events → *Diagnosis, Treatment, Recovery*
- Rules → *IF Fever AND Cough → Possible Flu*

This wide expressiveness allows the AI to understand the full medical scenario.

✔ 2. Unambiguous

Meaning:
Knowledge must have **one clear meaning**, not multiple interpretations.

Example:
Instead of vague text like:

- "Patient may have issues."

The AI uses precise logic:

- *Temperature(Patient)=102°F*
- *Symptom(Patient, Cough)=True*

This avoids confusion and ensures **consistent decisions**.

✔ 3. Efficient

Meaning:
Knowledge must be stored in a way that allows **fast reasoning**.

Example:
Instead of searching entire patient history, AI stores symptoms in indexed form:

- Fever: YES
- Cough: YES
- Oxygen: 93%

This allows fast rule checking:

IF Fever AND Cough AND Oxygen<94 THEN → Pneumonia Risk

Efficient structure → faster diagnosis.

✓ 4. Flexible

Meaning:
Easy to update, modify, or add new facts or rules.

Example:
If a new symptom of a disease is discovered:

Add rule:

IF High Fever AND Headache AND Red Eyes → Suspected Dengue

The system updates instantly **without redesigning everything**.

✓ 5. Structured

Meaning:
Knowledge must be organized meaningfully using:

- Graphs
- Frames
- Hierarchies
- Rules
- Ontologies

Example:
A medical knowledge structure:

```
Disease
 ├── Viral Disease
 │    ├── Flu
 │    ├── Dengue
 │    └── COVID
 │
```

└── Bacterial Disease

├── TB

└── Pneumonia

This tree helps AI reason systematically.

✔ 6. Supports Inference

Meaning:
AI should derive **new knowledge** from existing knowledge.

Example:
Facts:

- Fever
- Cough
- Low Oxygen

Rule:

- IF Fever AND Cough AND Low Oxygen → Pneumonia

Inference:
The AI concludes:
➡ *Patient is likely suffering from Pneumonia.*

⭐ Quick Summary Table

Feature	Meaning	Example in Medical AI
Expressive	Wide knowledge types	Symptoms, diseases, events
Unambiguous	Clear meaning	Exact symptom values
Efficient	Fast reasoning	Indexed symptom lookup
Flexible	Easy updates	Adding new disease rules
Structured	Organized form	Disease hierarchy tree
Supports Inference	Derives new facts	Diagnose pneumonia

Types of Knowledge

✅ (a) Inheritable Knowledge

Definition:

Knowledge passed or inherited through a **hierarchical structure**, like parent → child classes.

Example:

- Animal → Mammal → Dog
  If "All mammals breathe oxygen,"
  then **Dog inherits this knowledge**.

Used in AI:

Frames, semantic networks, ontologies.

✅ (b) Inferential Knowledge

Definition:

Knowledge that can be **derived using inference rules**.

Example:

Facts:

- "All humans are mortal"
- "Ram is human"
  Inference:
- "Ram is mortal"

Used in AI:

Inference engines, logical reasoning.

✅ (c) Relational Knowledge

Definition:

Knowledge that describes the **relationships between objects, events, or concepts**.

Example:

- "Sita is mother of Hari." (family relation)
- "Water is required for plants." (causal relation)
- "Doctor examines patient." (role relation)

Used in AI:

Graphs, knowledge graphs, semantic networks.

✅ (d) Heuristic Knowledge

Definition:

Knowledge gained through **experience**, used to **speed up problem solving**.
Not always perfect, but practical.

Example:

- “If a computer is slow, restart it first.”
- “In chess, control the center of the board.”
- “In hill-climbing, choose the highest-value move.”

Used in AI:

Heuristic search (A*, Greedy), expert systems.

✅ (e) Commonsense Knowledge

Definition:

Basic everyday knowledge that humans inherently know.

Examples:

- Water makes things wet
- Fire is hot
- People need food
- Objects fall when dropped

In AI:

Used in robots, NLP systems, planning systems.

✅ (f) Explicit Knowledge

(Already covered under declarative knowledge)
Knowledge that is written, formal, and structured.

Examples:

- Laws, rules
- Medical textbooks
- Databases

✅ (g) Uncertain Knowledge

Definition:

Knowledge where facts have probabilities, degrees of belief, or uncertainty.

Examples:

- “There is a 60% chance of rain.”
- “Patient may have malaria with probability 0.75.”
- Sensor reading confidence = 0.92

In AI:

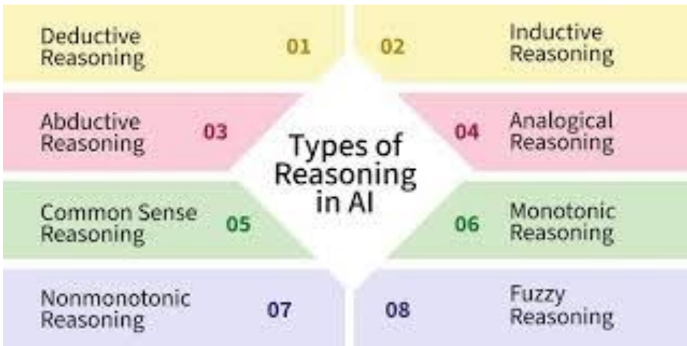
Bayesian networks, fuzzy logic, probabilistic reasoning.

⭐ Summary Table

Knowledge Type	Meaning	Example
Tacit / Procedural	Skill-based, cannot be fully written	Riding a bike, diagnosing from experience
Explicit / Declarative	Written, facts, easy to store	Definitions, rules, formulas
Inheritable	Passed down hierarchy	Dog inherits “Mammal breathes oxygen”
Inferential	Derived through logic	“Ram is mortal”
Relational	Relationships	Doctor–Patient, Parent–Child
Heuristic	Experience-based shortcuts	Chess rules, troubleshooting tips
Commonsense	General world knowledge	Fire is hot
Uncertain	Probabilistic knowledge	Weather forecast probabili

⭐ Types of Reasoning in AI

Artificial Intelligence uses different reasoning techniques to draw conclusions, make decisions, or infer new knowledge. The major types are:



1 Deductive Reasoning

Definition:

Reasoning from **general rules → specific conclusions**.
If the premises are true, the conclusion *must* be true.

Example:

- Rule: *All birds can fly*
- Fact: *Sparrow is a bird*
  ✓ Conclusion: *Sparrow can fly*

Used in: **Expert systems, Logic-based AI**

---

2 Inductive Reasoning

Definition:

Reasoning from **specific examples → general rule**.
Conclusion is probable, not guaranteed.

Example:

- Observation: *100 swans seen so far are white*
  ✓ Conclusion: *All swans are white* (may be false)

Used in: **Machine learning, Data-driven AI**

---

3 Abductive Reasoning

Definition:

Reasoning from **effects → possible causes**.
Often used for diagnosis.

Example:

- Fact: *Patient has fever + cough*
  ✓ Possible cause: *Flu infection*

Used in: **Medical diagnosis, Fault detection**

---

4 Analogical Reasoning

Definition:

Solving a new problem using similarity with an earlier known problem.

Example:

- "A battery drains in cold weather → similar to laptop battery issues"
  ✓ Apply same solution: *Keep device warm*

Used in: **Case-based reasoning, Similarity AI**

5 Common Sense Reasoning

Definition:

Reasoning based on everyday human knowledge.

Example:

- "If it is raining, the road will be wet."
- "Fire is hot."

Used in: **Robotics, NLP, Human-AI interaction**

---

6 Monotonic Reasoning

Definition:

Adding new knowledge **never invalidates previous conclusions**.
Conclusions only grow.

Example:

- Rule: "All mammals have lungs."
- Add: "Whale is a mammal."
  ✓ Conclusion remains valid.

Used in: **Classical logic, Mathematics**

---

7 Non-Monotonic Reasoning

Definition:

New information **can change or cancel old conclusions**.
More realistic for human reasoning.

Example:

- Rule: "Birds can fly."
- New info: "Penguins are birds but cannot fly."
  ✓ Old conclusion is revoked.

Used in: **Real-world AI, Planning, Blockchain logic**

---

8 Fuzzy Reasoning

Definition:

Reasoning with **degrees of truth** instead of strict true/false.
Useful for uncertain or vague data.

Example:

- "Temperature is warm (0.6 truth)"
- "Temperature is hot (0.3 truth)"

Used in: **Washing machines, AC systems, Fuzzy control**

⭐ Summary Table

Type of Reasoning	Meaning	Example
Deductive	General → Specific	All humans mortal → Ram mortal
Inductive	Specific → General	Many swans white → all white
Abductive	Effect → Possible cause	Fever → flu
Analogical	Similar cases	Solve new problem like old one
Common Sense	Human daily logic	Rain → wet road
Monotonic	New facts do not change old conclusions	Mammals → lungs
Non-Monotonic	New facts may change conclusions	Birds fly → penguin exception
Fuzzy	Degrees of truth	Warm = 0.6

⭐ Reasoning in Research:

How It Connects to Quantitative & Qualitative Approaches (NIC)

In research methodology, the two fundamental approaches are:

- **Deductive reasoning → Quantitative research**
- **Inductive reasoning → Qualitative research**

But other reasoning types like **abductive, analogical, fuzzy, non-monotonic**, etc., also play important roles in modern research designs.

⭐ 1 Deductive Reasoning in Research (Quantitative Approach)

Meaning (Research Context):
Start with **theory → hypothesis → data collection → testing**
You move from **general to specific**.

Example (Quantitative Research):
Theory: Exercise improves heart health.
Hypothesis: "Regular running reduces heart rate variability by 10%."
Test: Collect ECG data from 200 runners.
Conclusion: Hypothesis accepted or rejected.

Why it is used?
✓ Strong for testing established theories
✓ Works well with numbers, statistics
✓ Used in experiments, surveys

Deductive = the foundation of quantitative research.

⭐ 2 Inductive Reasoning in Research (Qualitative Approach)

Meaning:
Start with **observations → identify patterns → build theory**
Move from **specific to general**.

Example (Qualitative Research):
You conduct interviews with 30 remote workers and observe patterns:

- They feel isolated
- They work more hours
- Productivity varies

From these specific cases →
✓ You develop a theory about **remote work culture impacts**.

Why it is used?
✓ Good for exploring new phenomena
✓ Works with open-ended data (interviews, observations)
✓ Generates new theoretical frameworks

Inductive = core of qualitative research.

⭐ 3️⃣ Abductive Reasoning in Research (Mixed-Methods / Exploratory Studies)

Meaning:
Move from **incomplete observations → best possible explanation**.

Example in Research:
Patients from a village show unusual symptoms.
You don't have enough data, but:

- Many drank river water
- Symptoms resemble cholera

You **infer a likely cause** (contaminated water)
→ investigate further using mixed methods.

Used when:
✓ Data is incomplete
✓ Research is exploratory
✓ Hypotheses are created mid-study
This is increasingly important in **real-world applied research**.

⭐ 4️⃣ Analogical Reasoning in Research

Meaning:
Use results from one domain to understand another similar domain.

Example:
You study:

- "Mobile banking adoption in India"

Then compare it with:

- "Mobile banking adoption in Nepal"

You transfer insights because the contexts are similar.

Good for:

✓ Comparative studies
✓ Case-based reasoning
✓ Policy research

⭐ 5 **Common Sense Reasoning in Research**

Meaning:
Use everyday logic to interpret observations.

Example:
When studying traffic behavior:

- "More vehicles → more congestion"
- "Rain → slower movement"

These are not theories; they are **commonsense patterns** used to guide research.

Used to design:
✓ Interview questions
✓ Observation checklists
✓ Preliminary study frameworks

⭐ 6 **Monotonic Reasoning in Research**

Meaning:
New data does **not** change previous conclusions.

Example:
If decades of research show:

- Smoking increases lung cancer risk

Adding new samples will not change this fact.

Used when:
✓ Research is well-established
✓ Laws/theories are stable

⭐ 7 **Non-Monotonic Reasoning in Research**

Meaning:

New data **changes or contradicts** old conclusions.

Example:
Old belief:

- "Eggs increase cholesterol."

New research:

- "Eggs do not significantly affect cholesterol for most people."

Conclusion changed → **non-monotonic**.

Used in:
✓ Evolving scientific fields
✓ Behavior studies
✓ Medicine & public health

⭐ 8 **Fuzzy Reasoning in Research**
Meaning:
Used when variables are **not strictly true/false**, but have degrees.

Example:
Research on stress:
Stress = low, medium, high
(not binary)

Fuzzy logic in research helps in:
✓ Human behavior studies
✓ Risk modeling
✓ Environmental research

Issues in Knowledge Representation (KR)

These are the challenges an AI system faces when storing and using knowledge.

⭐ Issue 1: How to Represent Knowledge? (Representation Problem)

Should we use:

- Logic

- Semantic networks
- Frames
- Rules
- Ontologies
- Scripts

Choosing the right format affects system performance.

⭐ **Issue 2: Knowledge Acquisition (How to Get Knowledge?)**

Difficulties in:

- Extracting knowledge from experts
- Learning from data
- Converting real-world defects into symbolic form

Example: encoding a doctor's medical knowledge into rules.

⭐ **Issue 3: Knowledge Scope (What to Represent?)**

Should the system include:

- Facts
- Relationships
- Exceptions
- Time-based information

Defining boundaries is difficult.

⭐ **Issue 4: Ambiguity and Vagueness**

Natural language is ambiguous:

- "Bank" (river bank or financial bank?)
- "He is cold" (temperature or emotion?)

AI must avoid misinterpretation.

⭐ **Issue 5: Incomplete and Uncertain Knowledge**

Real-world knowledge is often:

- Uncertain

- Incomplete
- Noisy
- Probabilistic

AI must handle missing or uncertain data.

⭐ Issue 6: Updating Knowledge (Revision Problem)

Knowledge bases must be:

- **Flexible**
- **Easy to update**
- **Able to accept new facts without breaking old ones**

Example:
Earlier: “All birds fly”
Update: “Penguins are birds but do not fly.”

⭐ Issue 7: Efficient Storage and Retrieval

Knowledge must be stored in structures that allow:

- Fast search
- Efficient inference
- Low memory usage

Large knowledge bases (like Google Knowledge Graph) face this issue.

⭐ Issue 8: Reasoning and Inference Complexity

Logical reasoning may be:

- Computationally expensive
- Slow for large databases
- Involving heavy search

AI must balance reasoning accuracy and speed.

⭐ Issue 9: Multiple Representations for Same Knowledge

Example:
“Ram is Sita's brother”
Can be represented using:

- Semantic network
- Predicate logic
- Frame
- Ontology

Choosing one affects understanding and performance.

⭐ Issue 10: Representing Time and Change (Dynamic Knowledge)

Real-world knowledge changes:

- Stock prices
- Patient's health
- Weather
- Traffic

Representing evolving knowledge is challenging.

⭐ Knowledge Representation Systems

Knowledge Representation (KR) systems provide different ways to store, organize, and reason with knowledge in Artificial Intelligence.

Different KR models suit different types of knowledge.

---

1 Semantic Networks (Semantic Nets)

Definition:

A semantic network is a **graph-based** representation of knowledge, where

- **Nodes = concepts/objects**,
- **Edges = relationships** between concepts.

```
Animal
  |
(is-a)
  |
 Dog
  |
(has-property)
  |
Barks
```

Common Relationships:

- *Is-a* (inheritance)
- *Part-of*
- *Instance-of*
- *Has-property*

Features:

✓ Easy to visualize

✓ Supports inheritance
✓ Good for hierarchical knowledge
✓ Used in ontologies, WordNet

---

2 Frames

Definition:

Frames are **structured data objects** representing knowledge about a concept, object, or situation using **attributes (slots)** and their **values**.

Example (Car Frame):

```yaml
Frame: Car
  Slots:
    Color: Red
    Engine: 1500cc
    Fuel: Petrol
    Owner: Sanjeev
```

Structure:

- Frame = Concept
- Slots = Attributes
- Fillers = Values

Features:

✓ Very organized and hierarchical
✓ Supports defaults
✓ Supports inheritance
✓ Suitable for object-oriented representation

3 Conceptual Dependency (CD Theory)

Definition:

Introduced by *Roger Schank*, conceptual dependency represents the **meaning of sentences** in a language-independent, conceptual form.

Purpose:

✓ Avoid ambiguity
✓ Capture true meaning of actions

CD uses primitive actions, such as:

- **PTRANS** – physical transfer
- **ATRANS** – abstract transfer (give, receive)
- **MTRANS** – mental transfer (informing)
- **INGEST** – eating
- **EXPEL** – excretion

CD Representation:

Example Sentence:

"Ram gave Sita a book."

- ATRANS (possession transfer)
- From: Ram
- To: Sita
- Object: Book

4 Scripts

Definition:

Scripts represent **event sequences** or **stereotypical situations**.
Used for understanding stories, dialogues, and events.

Use-Cases:

✓ Story understanding
✓ Natural language processing
✓ Event prediction

Example:

Restaurant Script

```markdown
1. Enter restaurant
2. Sit at table
3. Order food
4. Eat food
5. Pay bill
6. Leave
```

---

5 Rule-Based Systems (Production Systems)

Definition:

Knowledge is represented as **IF–THEN rules**.

Structure:

- **IF** <condition>
- **THEN** <action>

Example:

```java
IF temperature > 101°F AND cough = true
THEN diagnose = Flu
```

Rules + Working Memory + Inference Engine = Production System

Features:

✓ Easy to update
✓ Human-readable
✓ Used in expert systems (MYCIN, DENDRAL)

Two Main Reasoning Methods:

- **Forward Chaining** (data-driven)
- **Backward Chaining** (goal-driven)

---

6 Propositional Logic (PL)

Definition:

Represents knowledge using **simple statements (propositions)** that are either **true or false**.

Example Propositions:

- P: "It is raining."
- Q: "The road is wet."

Compound Statements:

- AND (P ∧ Q)
- OR (P ∨ Q)
- NOT (¬P)
- Implication (P → Q)
- Biconditional (P ↔ Q)

Example:

"If it is raining, the road is wet":

```css
P → Q
```

Features:

✓ Simple
✓ Easy for computers
But ❌ cannot represent relations or quantifiers.

---

7 Predicate Logic (First-Order Logic, FOL)

Definition:

Extends propositional logic by allowing statements about **objects**, **properties**, and **relations**.

Syntax:

- **Predicates**: Loves(Ram, Sita)
- **Constants**: Ram, Nepal
- **Variables**: x, y
- **Functions**: FatherOf(x)
- **Quantifiers**:
  - ∀ (For all)
  - ∃ (There exists)

Example:

"All humans are mortal."

```css
∀x: Human(x) → Mortal(x)
```

"Ram is human."

```scss
Human(Ram)
```

Features:

✓ Powerful and expressive
✓ Represents relationships
✓ Supports inference

⭐ **Summary Table (Perfect for Exams)**

KR System	Representation Type	Example	Strength
Semantic Nets	Graph of nodes & links	Dog → Animal	Inheritance, hierarchical
Frames	Structured objects	Car: color, engine	Organized, default values
Conceptual Dependencies	Primitive conceptual acts	ATRANS, PTRANS	Natural language meaning
Scripts	Event sequences	Restaurant script	Story/event understanding
Rule-Based Systems	IF–THEN rules	IF fever THEN flu	Expert systems
Propositional Logic	True/false statements	P → Q	Simple logic
Predicate Logic	Objects & relations	Loves(Ram,Sita)	Very expressive

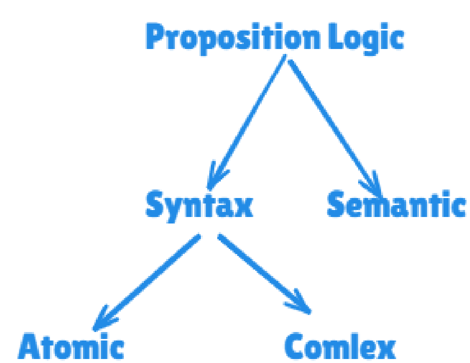
4.1 Propositional Logic and Its Resolution

Propositional Logic (PL) is a **formal logic system** used to represent and reason about **facts that are either true or false**.
It is the **foundation of logical reasoning in AI**, digital circuits, and mathematical logic.

👉 Each statement is called a **proposition** and has **only one truth value**: **True (T)** or **False (F)**.

Propositional Logic (PL) is the **simplest form of logic** used in AI to represent knowledge using **propositions (statements)** that are either:

TRUE, or **FALSE**



Examples of Propositions:

- **P:** "It is raining."
- **Q:** "The road is wet."
- **R:** "The light is ON."

In 1976, Robert Kowalski came up with an equation

Algorithm = Logic + Control

The logic component specifies the knowledge to be used in solving problems.

---

2. Syntax of Propositional Logic

Propositions can be combined using **logical connectives**:

Operator	Symbol	Meaning	
NOT	**¬P**	Negation	Today is not raining
AND	**P ∧ Q**	Logical conjunction	Sit in the table and write some niovel.
OR	**P ∨ Q**	Logical disjunction	You should drink or eat momo once at a time
Implication	**P → Q**	If P then Q	If you study earlier then there is chance of good grades
Biconditional	**P ↔ Q**	P if and only if Q	I ll go to barber shop iff I have to cut my hairs.

Syntax of Propositional Logic

Meaning : Syntax defines the **structure and grammar** of valid logical expressions.

Basic Elements

1. **Propositional symbols**:
   - P, Q (atomic propositions)

     Example: P = “It is raining” Q = “The ground is wet”
2. **Logical connectives:** ¬, ∧, ∨, →, ↔
3. **Parentheses**: Used to avoid ambiguity

Syntax Rules

- An atomic symbol is a valid formula
- If P is a formula, then ¬P is a formula
- If P and Q are formulae, then
  (P ∧ Q), (P ∨ Q), (P → Q), (P ↔ Q) are formulae

✅ Any expression following these rules is a **Well-Formed Formula (WFF)**.

Semantics of Propositional Logic

Semantics deals with the **meaning** of logical expressions by assigning **truth values**.

1 NOT Operator (Unary Operator)

Definition:
Negates (reverses) the truth value of a proposition.
Unary means it operates on **one proposition only**.

Symbol: ¬P or ~P

Example:

- P = "It is raining."
- ¬P = "It is NOT raining."

If P = True → ¬P = False.

Let:

- **P:** Ram is a boy
- **Q:** Boys drink juice
- **R:** Ram drinks juice

We will use **P, Q, R** for all operators.

P	¬P
T	F
F	T

✓ **Example (Story-based):**

- P = "Ram is a boy."
- ¬P = "Ram is NOT a boy."

If P = True → ¬P = False.

2 AND Operator (Conjunctive Operator)

Definition:
P ∧ Q is true **only if both P and Q are true**.

Symbol: ∧ (and)

Example:

- P = "It is sunny."
- Q = "It is warm."
- P ∧ Q = "It is sunny AND warm."

Truth table:

P	Q	P ∧ Q
T	T	T

Let:

- **P:** Ram is a boy
- **Q:** Boys drink juice
- **R:** Ram drinks juice

We will use **P, Q, R** for all operators.

P	Q	P ∧ Q
T	T	T
T	F	F
F	T	F
F	F	F

✓ **Example:**

- P = "Ram is a boy."
- Q = "Boys drink juice."
- P ∧ Q = "Ram is a boy AND boys drink juice."

Meaning → **Both statements are true.**

T	F	F
F	T	F
F	F	F

3 OR Operator (Disjunctive Operator)

Definition:
P ∨ Q is true **if at least one** of P or Q is true.

Symbol: ∨

Example:

- P = "I will study."
- Q = "I will play."
- P ∨ Q = "I will study OR play."

Truth table:

P	**Q**	**P ∨ Q**
T	T	**T**
T	F	**T**
F	T	**T**
F	F	F

Let:

- **P:** Ram is a boy
- **Q:** Boys drink juice
- **R:** Ram drinks juice

We will use **P, Q, R** for all operators.

P	**Q**	**P ∨ Q**
T	T	T
T	F	T
F	T	T
F	F	F

✓ Example:
P = "Ram is a boy."
Q = "Boys drink juice."

- P ∨ Q = "Ram is a boy OR boys drink juice."

Even if only one is true → whole statement = True.

4 IMPLICATION (Conditional Operator)

Definition:
P → Q means: *If P happens, then Q happens.*
Symbol: →

Example:

- P = "It rains."
- Q = "Road becomes wet."
- P → Q = "If it rains, then the road becomes wet."

Truth Table:

P	**Q**	**P → Q**
T	T	T
T	F	**F**
F	T	T
F	F	T

The only false case is **True → False.**

Let:

- **P:** Ram is a boy
- **Q:** Boys drink juice
- **R:** Ram drinks juice

We will use **P, Q, R** for all operators.

P	**Q**	**P → Q**
T	T	T
T	F	F
F	T	T
F	F	T

✓ Example (logical rule):

- P = "Ram is a boy."
- Q = "Ram drinks juice."
- P → Q = "If Ram is a boy, then Ram drinks juice."

This encodes the idea that **all boys drink juice**,
so *if Ram is a boy → he drinks juice.*

5 Precedence of Operators

When evaluating complex logical expressions, operator precedence is:

1. ¬ (NOT)
2. ∧ (AND)
3. ∨ (OR)
4. → (Implication)
5. ↔ (Biconditional)

Example: Evaluate:

$\neg P \lor Q \land R$

Step order:

1. $\neg P$
2. $Q \land R$
3. $\neg P \lor (Q \land R)$

⭐ 6 Tautology

Definition:
A proposition that is **always true**, for all truth values of variables.

Example:
$P \lor \neg P$
Always TRUE (law of excluded middle).

Let:

- **P:** Ram is a boy
- **Q:** Boys drink juice
- **R:** Ram drinks juice

We will use **P, Q, R** for all operators.

✓ Example:
$P \lor \neg P$

- "Ram is a boy OR Ram is NOT a boy."
  Always true — covers all possibilities.

⭐ 7 Contradiction

Definition:
A proposition that is **always false**.

Example:
$P \land \neg P$
Always FALSE.

Let:

- **P:** Ram is a boy
- **Q:** Boys drink juice
- **R:** Ram drinks juice

We will use **P, Q, R** for all operators.

✓ Example:
$P \land \neg P$

- “Ram is a boy AND Ram is NOT a boy.”
Impossible → Always false.

Validity

An argument is **valid** if the **conclusion is true whenever premises are true**.

Example (Modus Ponens)

1. P → Q
2. P
∴ Q

This argument is **valid**.

📌 Validity depends on **structure**, not real-world truth.

Topic: ***Medical Diagnosis (Simple Logic)***

1. P → Q : “If fever, then illness”
2. P : “Patient has fever”
∴ Q : “Patient is ill”

✓ Valid argument (Modus Ponens)

Well-Formed Formula (WFF)

A WFF is a **syntactically correct logical expression**.

Examples

✓ (P ∧ Q) → R
✓ ¬(P ∨ Q)

❌ P ∧ → Q (invalid syntax)
❌ (P Q ∧)

Topic: ***Library Management Rule***
Valid WFF:
(P ∧ Q) → R
Where:

- P: “Book issued”
- Q: “Due date passed”
- R: “Fine imposed”

📌 Syntax correctness ensures logical clarity.

Inference in Propositional Logic

Inference is the process of **deriving new conclusions** from known facts.

Common Inference Rules

- Modus Ponens
- Modus Tollens
- Resolution

Topic: ***Cybersecurity Alert***

1. P → Q : “If malware detected → system alert”
2. P : “Malware detected”
∴ Q : “System alert raised”

📌 Demonstrates logical reasoning.

Resolution

Resolution is a **rule of inference** used to prove statements by **contradiction**.

Resolution Rule
From:

- (P ∨ Q)
- (¬P ∨ R)

We can infer:

- (Q ∨ R)

Steps in Resolution

1. Convert formula to **Conjunctive Normal Form (CNF)**
2. Apply resolution rule
3. Derive empty clause (contradiction) → proof complete

Importance

- Core method in **automated theorem proving**

	• Used in **AI reasoning systems** **Topic:** ***Access Control System*** Clauses: • (¬P ∨ Q) → "If logged in, then access" • (P) → "User logged in" Resolving: → Q (Access granted) 📌 Used in **AI theorem proving & automated reasoning**.

Truth Table Properties of Propositional

Verifying a logical property by checking all possible truth values using a truth table.

---

1. NOT (¬) – Negation Property

Property

- Double negation law: $\neg(\neg P) = P$

Truth Table (Induction Proof)

P	¬P	¬(¬P)
T	F	T
F	T	F

✓ Since ¬(¬P) equals P in all cases, the property holds.

Example

- P: "Door is open"
- ¬¬P = "Door is open"

---

2. AND (∧) – Conjunction Properties

(a) Identity Law $P \wedge T = P$

P	T	P ∧ T
T	T	T
F	T	F

✓ Matches P → property proven

(b) Domination Law $P \wedge F = F$

P	F	P ∧ F
T	F	F
F	F	F

✓ Always false

Example

- P: "Attendance ≥ 80%"
- Rule: Attendance ∧ True = Attendance

---

3. OR (∨) – Disjunction Properties

(a) Identity Law $P \vee F = P$

P	F	P ∨ F
T	F	T
F	F	F

✓ Matches P

(b) Domination Law $P \vee T = T$

P	T	P ∨ T
T	T	T
F	T	T

✓ Always true

Example

- P: "High GPA"
- High GPA OR True → Always eligible

---

4. Idempotent Laws

Property

$$P \land P = P; P \lor P = P$$

P	P ∧ P	P ∨ P
T	T	T
F	F	F

✓ Both expressions reduce to P

Example

- "Login valid AND login valid" → Login valid

---

5. Commutative Laws

Property

$$P \land Q = Q \land P$$
$$P \lor Q = Q \lor P$$

P	Q	P ∧ Q	Q ∧ P	P ∨ Q	Q ∨ P
T	T	T	T	T	T
T	F	F	F	T	T

P	Q	P ∧ Q	Q ∧ P	P ∨ Q	Q ∨ P
F	T	F	F	T	T
F	F	F	F	F	F

✓ Columns are identical → property proven

Example

- Exam ∧ Assignment = Assignment ∧ Exam

---

6. Associative Laws

Property $$(P \land Q) \land R = P \land (Q \land R)$$

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

✓ Both sides equal in all rows

Example

- (Marks ∧ Attendance) ∧ FeePaid

---

7. Distributive Laws

Property $$P \land (Q \lor R) = (P \land Q) \lor (P \land R)$$

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

P	Q	R	QVR	P∧(QVR)	(P∧Q)V(P∧R)
F	T	T	T	F	F

✓ Both expressions match → proven

Example

- Eligible AND (Merit OR Quota)

---

8. Implication (→) Property

Property $$P \to Q \equiv \neg P \lor Q$$

P	Q	P→Q	¬P	¬PVQ
T	T	T	F	T
T	F	F	F	F
F	T	T	T	T
F	F	T	T	T

✓ Columns identical → equivalence proven

Example

- If fever → illness
- Equivalent: No fever OR illness

---

9. Biconditional (↔) Property

Property $$P \leftrightarrow Q \equiv (P \to Q) \land (Q \to P)$$

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

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

✓ Equivalence holds

Example

- Login success ↔ Correct credentials

---

10. Tautology Property

Property $$P \vee \neg P = T$$

P	¬P	P∨¬P
T	F	T
F	T	T

✓ Always true → tautology

Example

- System is ON or NOT ON

---

11. Contradiction Property

Property $$P \wedge \neg P = F$$

P	¬P	P∧¬P
T	F	F
F	T	F

✓ Always false

Example

- Door open AND door not open

---

12. De Morgan's Laws

Property $\neg(P \land Q) = \neg P \lor \neg Q$

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

✓ Property verified by induction

Example

- NOT (Passed AND PaidFee)

Key Properties of Propositional Logic

- Binary truth values (T/F)
- Decidable logic
- Sound and complete
- Limited expressiveness (no variables or quantifiers)

Limitations of PL

- Cannot represent **objects or relations**
- Cannot handle **quantifiers**
- Not suitable for complex real-world knowledge

👉 These limitations lead to **Predicate Logic**.

Numerical

Given Propositions

P: It is raining **Q**: The road is wet **R**: Traffic is slow **S**: I reach office late

Ans

Q#	Statement	Logic Form / Result
Q1	"If it is raining, then the road is wet."	**$P \to Q$**
Q2	"It is raining and traffic is slow."	**$P \land R$**
Q3	"The road is wet or traffic is slow."	**$Q \lor R$**
Q4	"I reach office late iff traffic is slow."	**$S \leftrightarrow R$** (same as $R \leftrightarrow S$)
Q5	Find truth of $(P \land Q)$ when P=T, Q=F	$T \land F =$ **False**
Q6	Construct truth table for $\neg P$	see table below
Q7	$P \lor \neg P$ is?	**Tautology**
Q8	Validity of: $P \to Q, P \therefore Q$	**Valid** (Modus Ponens)
Q9	Convert $R \to S$ using only $\neg$ and $\lor$	**$\neg R \lor S$**
Q10	From $(P \land R) \to S$ and $P \land R$, derive S	**S** (Modus Ponens)

Q6 Truth table for ¬P (table only)

P	**¬P**
T	F
F	T

Que 2 / CW/HW

Let: P: "Exam is tomorrow" Q: "I will study tonight" R: "I will pass the exam" S: "I will celebrate"

Questions

1. Translate: "If I study tonight, then I will pass the exam."
2. Translate: "Exam is tomorrow and I will study tonight."
3. Translate: "If I pass the exam, then I will celebrate."
4. Translate: "I will celebrate iff I pass the exam."
5. Write using symbols: "Exam is not tomorrow."
6. Build WFF for: "If exam is tomorrow, then (I study tonight and I will pass)."
7. Decide type: Is $P \lor \neg P$ tautology/contradiction/contingency?
8. Decide type: Is $P \land \neg P$ tautology/contradiction/contingency?
9. Check validity: $P \rightarrow R, Q \rightarrow R, (P \lor Q) \Rightarrow R$
10. Convert: $(P \rightarrow Q)$ into only $\neg$ **and** $\lor$

Predicate Logic (First-Order Predicate Logic – FOPL)

1. What is Predicate Logic?

Predicate Logic is an extension of **Propositional Logic** that allows us to represent:

- **Objects**
- **Properties of objects**
- **Relations among objects**
- **Quantification (for all / there exists)**

👉 It is much more **expressive** and suitable for **real-world knowledge representation**.

---

2. Why Predicate Logic is Needed

Propositional Logic cannot express:

- "All students passed the exam"

- “Some patients have fever”
- “Every human is mortal”

Predicate Logic solves this by introducing **predicates, variables, and quantifiers**.

---

3. Basic Components of Predicate Logic

3.1 Constants

Represent specific objects.

- Examples: Ram, Sita, Kathmandu, Car1

3.2 Variables

Represent arbitrary objects.

- Examples: x, y, z

3.3 Predicates

Represent **properties or relations**.

- Student(x) → x is a student
- Teaches(x, y) → x teaches y
- Greater(x, y) → x is greater than y

3.4 Functions

Map objects to objects.

- father(x)
- age(x)

Example:

age(Ram) = 20

---

4. Syntax of Predicate Logic

Syntax defines **how valid formulas are written**.

Atomic Formula

A predicate applied to terms:

Student(Ram)

Teaches(Teacher, AI)

Compound Formula

Formed using logical connectives:

Student(x) ∧ Intelligent(x)

Rules

- If P is a formula → ¬P is a formula
- If P and Q are formulae → (P ∧ Q), (P ∨ Q), (P → Q) are formulae

✓ Any correctly formed expression is a **Well-Formed Formula (WFF)**.

---

5. Semantics of Predicate Logic

Semantics gives **meaning** to formulas.

It depends on:

- **Domain of discourse** (set of objects)
- **Interpretation** of predicates and functions
- **Assignment of variables**

Example:

Domain = {Ram, Sita}

Student(Ram) = True

Student(Sita) = True

---

6. Quantifiers

6.1 Universal Quantifier (∀)

Means "for all".

Example:

∀x Human(x) → Mortal(x)

Meaning: *All humans are mortal.*

6.2 Existential Quantifier (∃)

Means "there exists".

Example:

∃x Student(x) ∧ Intelligent(x)

Meaning: *There exists a student who is intelligent.*

---

7. Free and Bound Variables

- **Bound variable**: inside quantifier

  ∀x Student(x)

- **Free variable**: not quantified

  Student(x)

📌 A formula with no free variables is called a **closed formula**.

---

8. Inference in Predicate Logic

8.1 Universal Instantiation (UI)

From:

∀x P(x)

Infer:

P(Ram)

8.2 Existential Instantiation (EI)

From:

∃x P(x)

Infer:

P(c) (c is a new constant)

8.3 Modus Ponens

From:

P → Q

P

Infer:

Q

---

9. Resolution in Predicate Logic

Resolution is a **mechanical inference method** used in AI.

Steps

1. Convert formulas to **Clause Form**
2. Remove quantifiers (Skolemization)
3. Apply **Unification**
4. Apply **Resolution Rule**

Example

Clauses:

$(\neg Human(x) \lor Mortal(x))$

$Human(Ram)$

Resolve ⇒

$Mortal(Ram)$

---

10. Example (Real-World Representation)

Statement

"All students who study AI are intelligent. Ram studies AI."

Predicate Logic

$\forall x\ (StudiesAI(x) \rightarrow Intelligent(x))$

$StudiesAI(Ram)$

Inference

$Intelligent(Ram)$

---

11. Properties of Predicate Logic

- More expressive than propositional logic
- Supports quantification
- Uses variables and relations
- Semi-decidable (inference may not always terminate)

---

12. Limitations

- More complex reasoning

- Computationally expensive
- Not suitable for vague concepts (handled by fuzzy logic)

13. Predicate Logic vs Propositional Logic

Feature	Propositional Logic	Predicate Logic
Objects	❌ No	✅ Yes
Quantifiers	❌ No	✅ Yes
Relations	❌ No	✅ Yes
Expressiveness	Low	High
Complexity	Low	Higher

14. Applications of Predicate Logic

- Expert systems
- Knowledge bases
- AI reasoning engines
- Natural language understanding
- Database query systems

Case Study: Smart University Academic Decision System

A large public university has implemented an **AI-driven Academic Decision Support System** to assist the examination department, faculty advisors, and scholarship committee.
The system uses **First Order Predicate Logic (FOPL)** to represent academic rules, student conditions, and institutional policies so that decisions are **consistent, transparent, and logically provable**.

The university follows these academic policies:

- Every **student** must maintain **minimum attendance** to be eligible for examinations.
- A student who has **attendance ≥ 80%** and has **submitted all assignments** is considered **exam eligible**.
- Any student who is **exam eligible** and has a **GPA ≥ 3.6** qualifies for a **merit scholarship**.
- A student who has **failed a subject** must attend **remedial classes**.
- Any student who attends remedial classes and passes the re-exam is considered to have **cleared the subject**.
- Students who clear all subjects and complete the final project are eligible for **degree award**.

The system stores real-time data from attendance records, assignment submissions, exam results, and project evaluations.

The following facts are currently stored in the system:

- Ram is a registered student of the university.
- Ram has an attendance of **85%**.
- Ram has submitted **all required assignments**.
- Ram has achieved a **GPA of 3.8**.
- Ram has **passed all subjects**.
- Ram has **successfully completed the final project**.

The university wants the AI system to determine, using **predicate logic inference**, whether **Ram qualifies for a merit scholarship and degree award**.

Predicate Vocabulary (Given)

Constants

- Ram

Predicates

- Student(x) – x is a student
- Attendance80(x) – x has attendance ≥ 80%
- SubmittedAllAssignments(x) – x submitted all assignments
- ExamEligible(x) – x is eligible for exam
- GPAHigh(x) – x has GPA ≥ 3.6
- MeritScholarship(x) – x qualifies for merit scholarship
- FailedSubject(x) – x has failed a subject
- RemedialClass(x) – x attends remedial classes
- ClearedSubject(x) – x has cleared subject
- CompletedProject(x) – x completed final project
- DegreeEligible(x) – x is eligible for degree award

Knowledge Base (First Order Predicate Logic Rules)

1. $\forall x\ [(Student(x) \wedge Attendance80(x) \wedge SubmittedAllAssignments(x)) \rightarrow ExamEligible(x)]$
2. $\forall x\ [(ExamEligible(x) \wedge GPAHigh(x)) \rightarrow MeritScholarship(x)]$
3. $\forall x\ [FailedSubject(x) \rightarrow RemedialClass(x)]$

4. $\forall x\ [(RemedialClass(x) \land PassedReExam(x)) \rightarrow ClearedSubject(x)]$
5. $\forall x\ [(ClearedSubject(x) \land CompletedProject(x)) \rightarrow DegreeEligible(x)]$

Facts (Ground Predicates)

6. $Student(Ram)$
7. $Attendance80(Ram)$
8. $SubmittedAllAssignments(Ram)$
9. $GPAHigh(Ram)$
10. $ClearedSubject(Ram)$
11. $CompletedProject(Ram)$

- Every **student** must maintain **minimum attendance** to be eligible for examinations.
- A student who has **attendance ≥ 80%** and has **submitted all assignments** is considered **exam eligible**.
- Any student who is **exam eligible** and has a **GPA ≥ 3.6** qualifies for a **merit scholarship**.
- A student who has **failed a subject** must attend **remedial classes**.
- Any student who attends remedial classes and passes the re-exam is considered to have **cleared the subject**.
- Students who clear all subjects and complete the final project are eligible for **degree award**.

4.2 Predicate Logic and Its Resolution

4.3 Unification Algorithm

4.4 Forward Chaining, Backward Chaining and Conflict Resolution