Lecture Notes No. 9			
Topic:	**Spatial Reasoning & Expert Systems**	**Week No.**	13-14
Course Code:	**CSST101**	**Term:**	1st Semester
Course Title:	**Advance Knowledge Representation and Reasoning**	**Academic Year:**	2025-2026
Student Name		**Section**	
Due date		**Points**	

Learning Outcomes

By the end of this lesson, you should be able to:

1. Explain the fundamentals of spatial reasoning and spatial representations.
2. Apply spatial logic using adjacency and region relations.
3. Understand expert systems and their components.
4. Define domain rules and encode reasoning tasks in Python.

1. What is Spatial Reasoning?

Spatial reasoning allows AI to **understand and reason about space, positions, and relations between objects**.

Example:

- "Room A is next to Room B."
- "Object X is north of Object Y."
- AI can answer queries like "Which rooms are reachable from Room A?"

Definition:

Spatial reasoning is the ability to **model, represent, and infer relationships** between entities in a space.

2. Key Concepts

Component	Description	Example
Grid Worlds	Discrete spaces represented as grids	Rooms in a building
Adjacency	Objects that are next to each other	adjacent(RoomA, RoomB)
RCC (Region Connection Calculus)	Represents topological relations	disconnected(A, B)
Expert System	AI system using rules to make decisions	Medical diagnosis system

Key Idea:

- Spatial reasoning helps AI navigate, plan, and analyze environments.
- Expert systems **encode domain knowledge as rules** and make automated decisions.

3. Spatial Relations in Python

Example: Using adjacency in a grid:

```
adjacency = {
    "RoomA": ["RoomB", "RoomC"],
    "RoomB": ["RoomA", "RoomD"],
    "RoomC": ["RoomA"],
    "RoomD": ["RoomB"]
}

def are_adjacent(room1, room2):
    return room2 in adjacency.get(room1, [])

print(are_adjacent("RoomA", "RoomB"))  # True
print(are_adjacent("RoomC", "RoomD"))  # False
```

Exercise:

- Add more rooms and objects.
- Write queries for reachable locations or neighbors.

4. Expert Systems

An expert system is an AI program that **uses rules to reason about a specific domain**.

Components:

1. **Knowledge Base:** Facts and rules about the domain.
2. **Inference Engine:** Applies rules to derive conclusions.
3. **Working Memory:** Stores current facts and state.

Example:

- Rule: "If patient has fever and cough → suspect flu."
- Knowledge base contains symptoms and relationships.
- System infers possible diagnoses based on user input.

5. Lab Activity (Python)

Goal: Implement spatial queries and simple expert system rules.

Sample Tasks:

- Encode rooms and adjacency relations.
- Implement rule-based queries for navigation or decision-making.
- Group Activity: Define rules for a chosen domain, e.g., smart building, medical diagnosis, or traffic system.

Challenge:

- Add reasoning for indirect relations, e.g., "If Room A is adjacent to Room B and Room B to Room D, can we reach Room D from Room A?"

6. Applications in AI

Field	Example of Use
Robotics	Path planning and navigation
Geographic Information Systems	Analyzing regions and spatial patterns
Smart Homes	Controlling devices based on locations
Expert Systems	Diagnostics, planning, and decision support

7. Reflection and Discussion

1. How does spatial reasoning differ from temporal reasoning?
2. Can you think of real-life situations where adjacency or topological relations matter?
3. Why are expert systems still relevant in certain AI applications?

8. Summary

- Spatial reasoning models **positions, adjacency, and topological relations**.
- RCC provides formal ways to reason about connected regions.
- Expert systems use **knowledge bases and rules** to make decisions.
- Python can encode spatial relations and rule-based reasoning for practical simulations.

Self-Check

1. What is the purpose of adjacency relations in AI?
2. How does an expert system use rules to infer conclusions?
3. How can Python help implement spatial reasoning tasks?
4. Give an example of a domain where expert systems are useful.