Lecture Notes No. 11			
Topic:	**Semantic Networks & Ontologies**	**Week No.**	16
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 concepts of semantic networks and ontologies.
2. Represent relationships between concepts using network structures.
3. Encode semantic networks in Python.
4. Apply reasoning to perform knowledge base tasks using networks and ontologies.

1. What are Semantic Networks?

Semantic networks are **graph structures** used to represent knowledge in terms of **nodes (concepts) and edges (relationships)**.

Example:

- Node: "Bird"
- Node: "Can Fly"
- Edge: "Bird → can → Fly"

Definition:

Semantic networks encode knowledge visually and formally to **show relationships between entities**, enabling AI systems to reason about them.

2. What are Ontologies?

Ontologies define **formal, structured representations of a domain**, specifying:

- **Concepts** (classes)
- **Relationships** (properties)
- **Constraints** (rules for valid relations)

Example:

- Class: Bird
- Subclass: Penguin
- Relation: canFly → True/False depending on subclass

Key Idea:

- Ontologies allow AI to understand **hierarchies, inheritance, and domain constraints**.
- Useful for semantic reasoning, interoperability, and knowledge sharing.

3. Representing Semantic Networks in Python

Example: Using dictionaries for nodes and edges

```python
semantic_network = {
    "Bird": {"canFly": True, "hasFeathers": True},
    "Penguin": {"isA": "Bird", "canFly": False},
    "Sparrow": {"isA": "Bird", "canFly": True}
}

def can_fly(animal):
    return semantic_network.get(animal, {}).get("canFly", None)

print(can_fly("Penguin"))  # Output: False
print(can_fly("Sparrow"))  # Output: True
```

Exercise:

- Add more animals and properties.
- Query inheritance relationships, e.g., which birds can fly.

4. Knowledge Base Tasks

1. **Inheritance Reasoning:** Subclasses inherit properties from superclasses unless overridden.
2. **Property Queries:** Check if a property holds for a given concept.
3. **Consistency Checks:** Ensure network/ontology rules are respected.

Example Query:

- "Can a penguin fly?" → False (overrides Bird property)
- "Does a sparrow have feathers?" → True (inherited from Bird)

5. Lab Activity (Python)

Goal: Encode a semantic network and perform ontology reasoning.

Sample Tasks:

- Define concepts and subclasses with properties.
- Implement Python functions for querying relationships and inherited properties.
- Test reasoning over multiple levels of hierarchy.

Challenge:

- Handle conflicting properties and exceptions, e.g., "Flightless birds."

6. Applications in AI

Field	Example of Use
Natural Language Processing	Represent word meanings and relationships
Knowledge Management	Organize domain knowledge for reasoning
Semantic Web	Enable machines to understand web data
Intelligent Agents	Reason about domain concepts and relations

7. Reflection and Discussion

1. How do semantic networks help AI represent complex knowledge?
2. How do ontologies improve reasoning and consistency?
3. Can you give a real-world scenario where semantic networks would be useful?

8. Summary

- Semantic networks represent knowledge using **nodes and edges**.
- Ontologies define **structured, hierarchical domain knowledge**.
- Python can encode and query networks to perform reasoning tasks.
- Useful in NLP, knowledge management, semantic web, and intelligent agents.

Self-Check

1. What is the difference between a semantic network and an ontology?
2. How does inheritance work in ontologies?
3. How can Python simulate semantic network reasoning?
4. Give an example of a domain suitable for ontological reasoning.