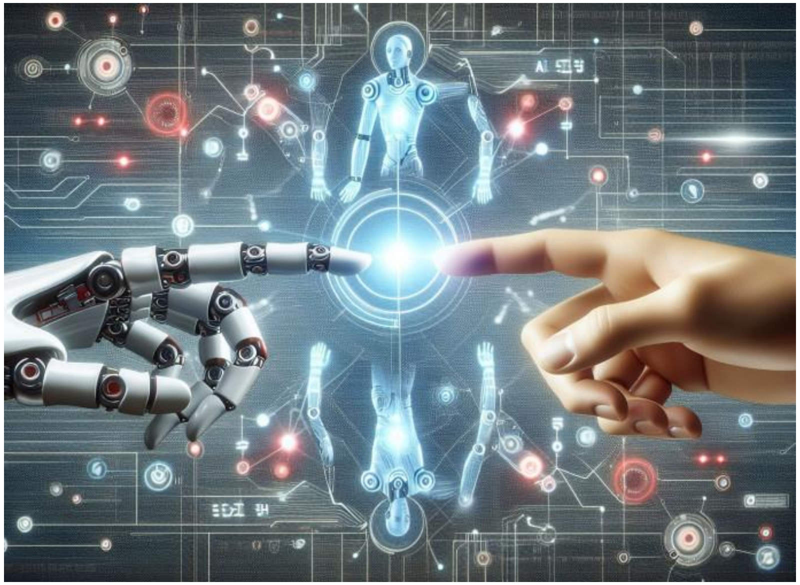
Lecture 4
Expert systems



Dr. Fatma Eskander
Math.department, Faculty of Science, Mansoura University

Knowledge Representation Techniques

Knowledge Representation Techniques

Common Techniques:

1- Productions (Rules)

2- Semantic Networks

3- Frames

4- Logic

Object-Attribute-Value Triple (O-A-V)

Definition: A way to represent facts as three-part statements

Format: (Object, Attribute, Value)

- **Object:** The entity (e.g., My Car)
- **Attribute:** The property (e.g., Color)
- **Value:** The property's value (e.g., Red)

Examples:

(Student, Age, 20)

(Apple, Taste, Sweet)

(House, Size, Large)

(Book, Author, "Shakespeare")

3- FRAMES

Definition: A data structure that represents a stereotypical object, event, or concept. It's like a form with pre-defined slots to be filled.

Structure:

- Frame Name (concept)
- Slots (attributes)
- Fillers (values for attributes)
- Default values
- Procedures (methods)

In OAV terms, the car is the object, the slot name is the attribute, and the fillers is the value.

3- FRAMES

Examples:

Slots	Fillers
manufacturer	General Motors
model	Chevrolet Caprice
year	1979
transmission	automatic
engine	gasoline
tires	4
color	blue

3- FRAMES

Examples:

Frame: PERSON

- Name: [John Smith]
- Age: [30]
- Occupation: [Engineer]
- Address: [123 Main St]
- Default Height: [170 cm]

Frame: CAR

- Make: [Toyota]
- Model: [Camry]
- Year: [2020]
- Color: [Blue]
- Has: [4 wheels]
- Can: [Drive, Transport]

Semantic nets provide 2-dimensional knowledge (node-link); frames provide 3-dimensional knowledge (object-attribute-value).

4- Logic and Sets

Logic: A formal system for reasoning and deriving conclusions

Sets: Collections of distinct objects

Purpose in Knowledge Representation:

- Provide unambiguous representation
- Enable automated reasoning
- Support mathematical proof of conclusions

Applications: Expert systems, theorem proving, database queries

4- Logic and Sets

Forms of Logic

A-Propositional Logic: Deals with simple propositions (statements that are true or false) and their connectives (AND, OR, NOT, etc.).

B- Predicate Logic (First-Order Logic): More powerful. Deals with predicates (properties of objects) and quantifiers (∀ "for all", ∃ "there exists").

4- Logic and Sets

Venn Diagrams

Visual representations of logical relationships between sets using overlapping circles.

Uses:

Show intersections between sets

Illustrate logical operations (union, intersection, complement)

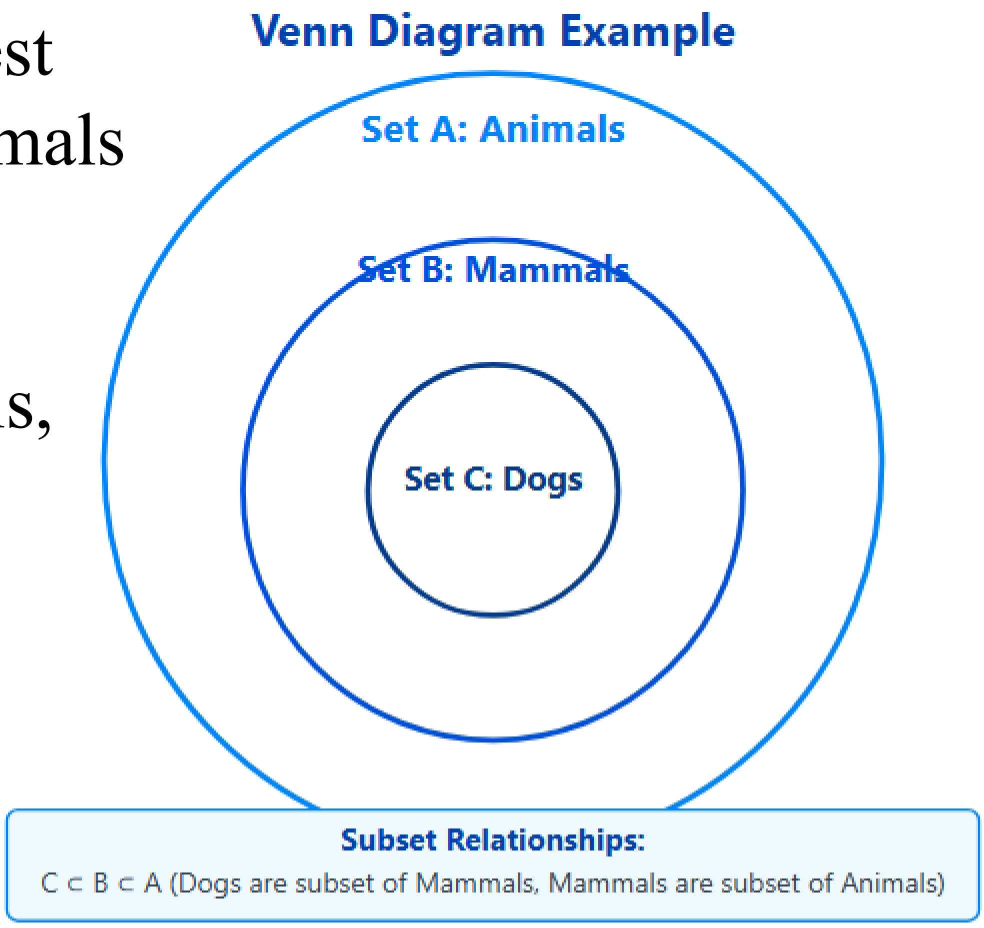
Visualize categorical relationships

4- Logic and Sets

- **Set A (Animals)** - The largest blue circle containing all animals

- **Set B (Mammals)** - A medium circle inside Animals, showing mammals are a subset of animals

- **Set C (Dogs)** - The smallest circle inside Mammals, showing dogs are a subset of mammals



4- Logic and Sets

A-Propositional Logic:

A formal system dealing with propositions and their relationships

Propositions: Statements that are either TRUE or FALSE

Examples of Propositions:

P: "It is raining" (True or False)

Q: "The ground is wet" (True or False)

R: "I will use an umbrella" (True or False)

Propositional Logic

A-Propositional Logic:

Logical Connectives:

- AND (∧): $P \wedge Q$
- OR (∨): $P \vee Q$
- NOT (¬): $\neg P$
- IMPLIES (→): $P \rightarrow Q$
- IF AND ONLY IF (↔): $P \leftrightarrow Q$

Truth Tables

Definition: Tables showing all possible truth values for logical expressions

AND (∧) Truth Table:

P	Q	P ∧ Q
T	T	T
T	F	F
F	T	F
F	F	F

Truth Tables

OR (V) Truth Table:

P	Q	P V Q
T	T	T
T	F	T
F	T	T
F	F	F

NOT (¬) Truth Table:

P	¬P
T	F
F	T

Truth Tables

IMPLIES (→) Truth Table:

P	Q	$P \rightarrow Q$
T	T	T
T	F	F
F	T	T
F	F	T

Note: Implication is only false when P is true and Q is false

Knowledge vs. Expert Systems

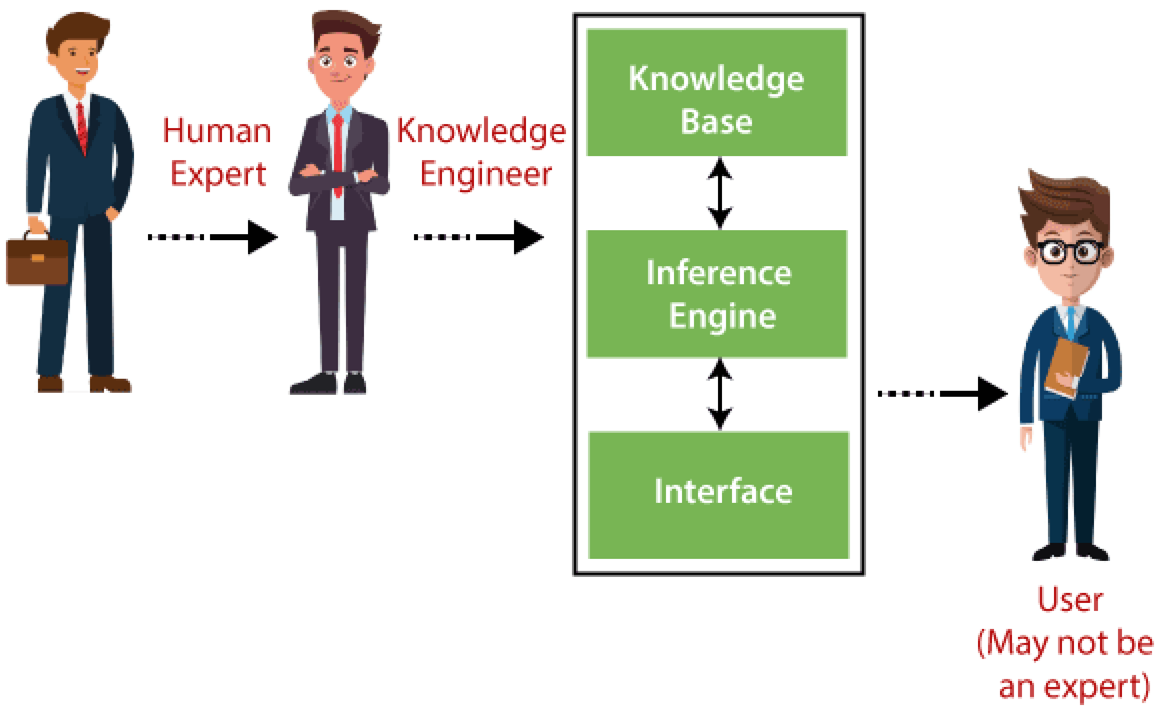
Knowledge representation is key to the success of expert systems.

Expert systems are designed for knowledge representation based on rules of logic called inferences.

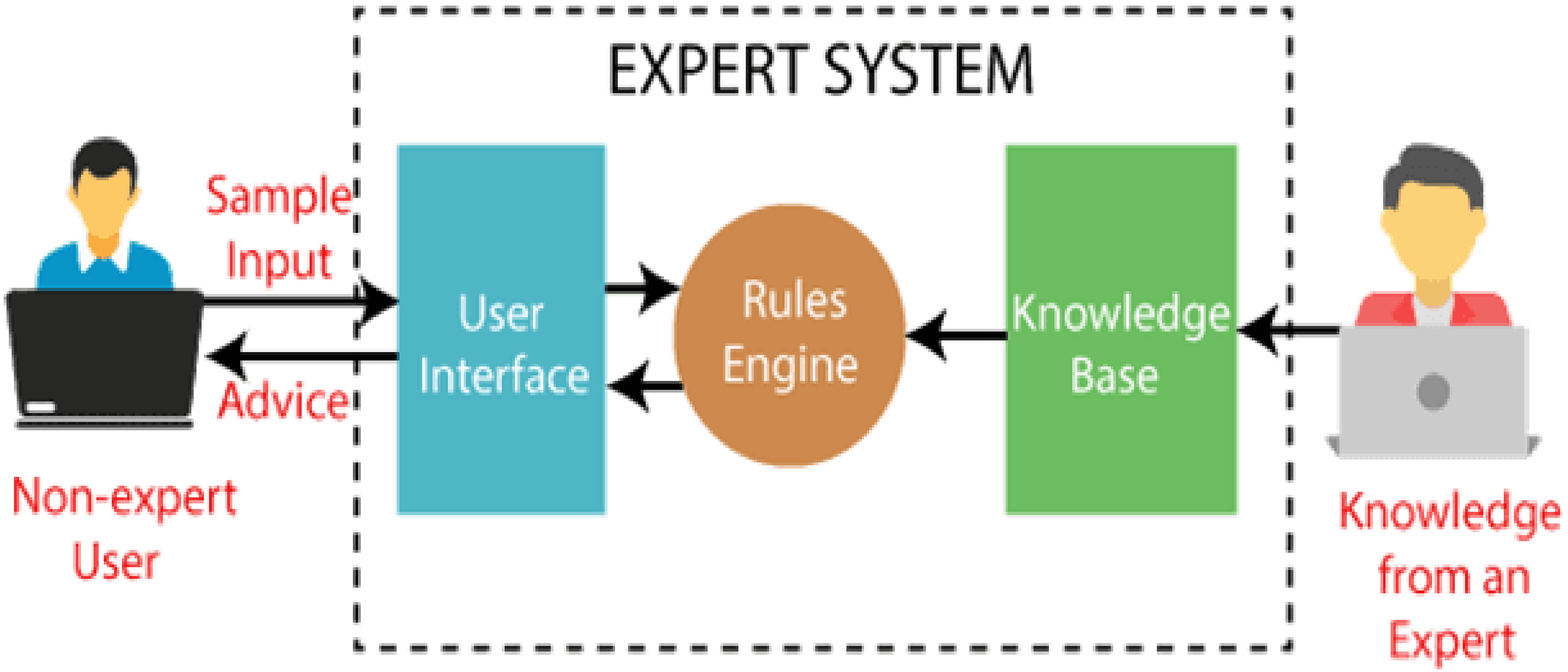
KEY DIFFERENCES

Aspect	Knowledge	Expert Systems
Definition	Information, understanding, and skills acquired through experience or education	Computer programs that use knowledge to solve problems like human experts

Aspect	Knowledge	Expert Systems
Storage	Human brain, documents, databases	Computer knowledge base
Scope	Broad, general	Narrow, specialized
Application	Any situation	Specific domain problems



Methods of Inference



Foundational Structures

What is a Graph?

•**Definition:** A graph G = (V, E) is a mathematical structure consisting of:

- •**V (Nodes):** A set of entities.
- •**E (Edges):** A set of connections between pairs of vertices.

•**Visual:** A simple diagram with circles (nodes) and connecting lines (edges).

Foundational Structures

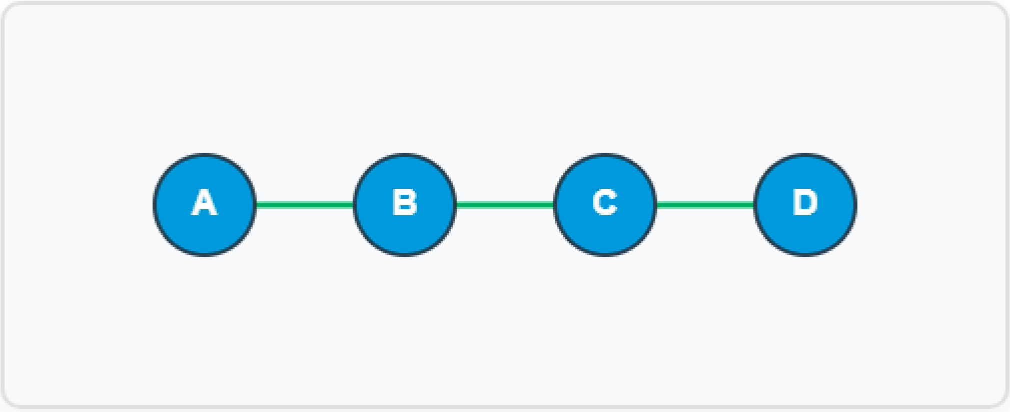
Features of graph

1. Path

A **path** is a sequence of edges that connects a sequence of distinct **nodes**.

A- Simple Path: No repeated nodes

Example: A → B → C → D

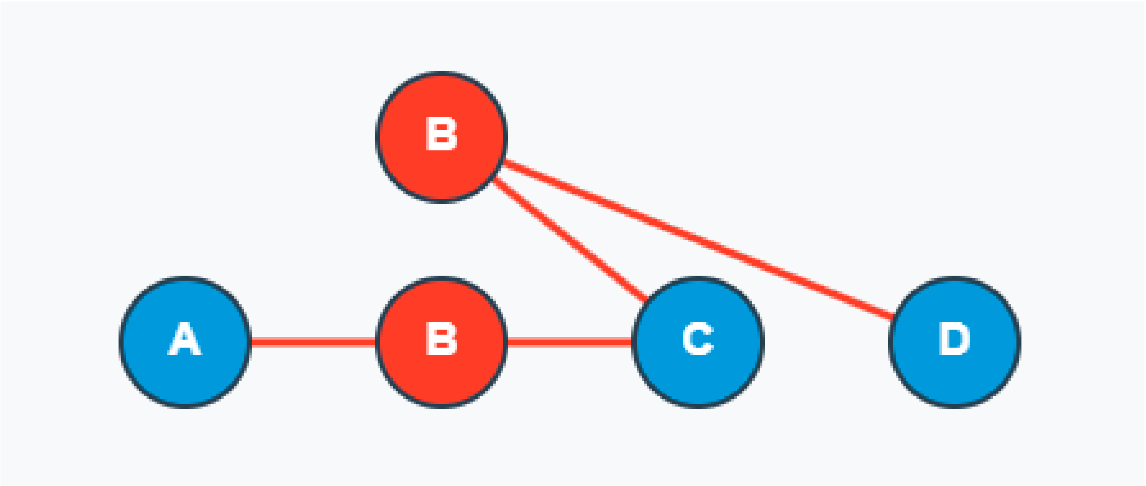


Foundational Structures

Features of graph

B-Non-simple Path: Has repeated nodes.

Example: A → B → C → B → D



Foundational Structures

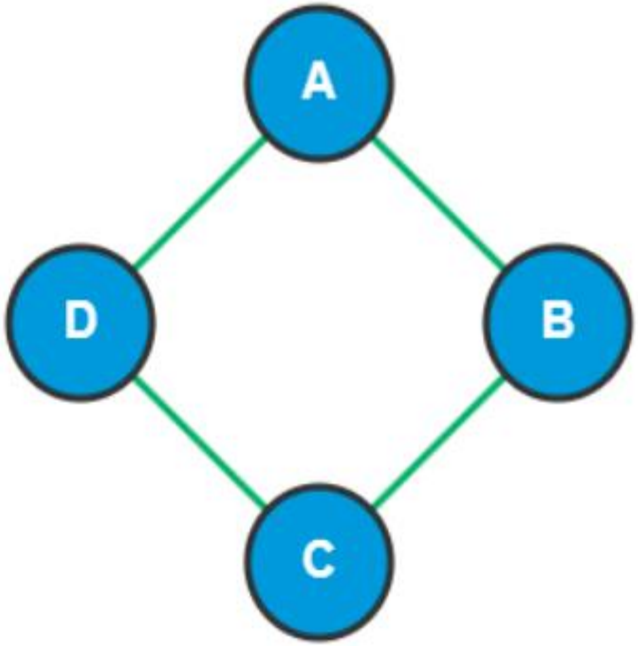
Features of graph

2. Circuit (Cycle)

A path that starts and ends at the same node.

A- Simple Circuit: Visits each node exactly once before returning to start.

Example: A → B → C → D → A



2. Circuit (Cycle)

B- Eulerian Circuit

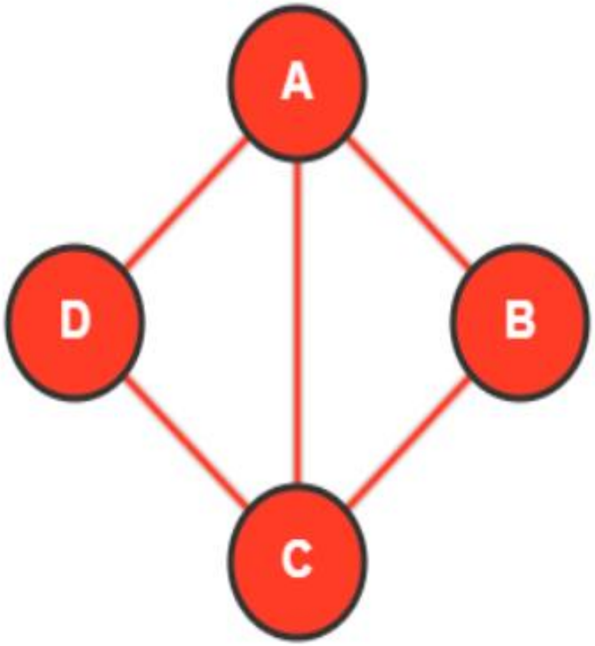
Visits every **edge** exactly once and starts and ends at the same node.

A graph has an Eulerian circuit if and only if all of its nodes have an **even degree**.

Node Degrees (Number of edges connected to each node)

B- Eulerian Circuit

Ex (1):



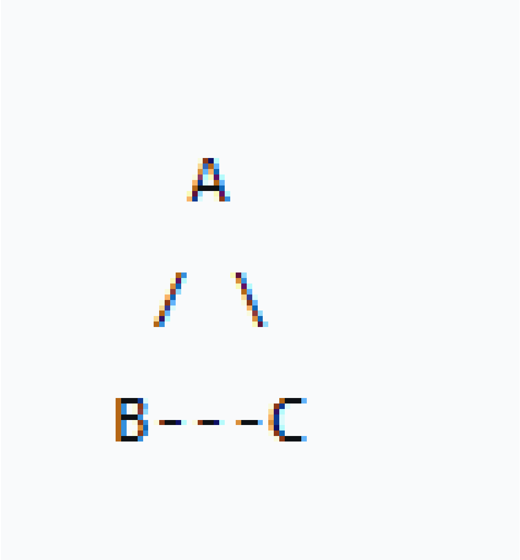
Degree Analysis:

A: Connected to B, D, C → **Degree = 3** (odd)
B: Connected to A, C → **Degree = 2** (even)
C: Connected to B, D, A → **Degree = 3** (odd)
D: Connected to C, A→ **Degree = 2** (even)

❌ **This graph has NO Eulerian circuit** because nodes A and C have odd degrees.

B- Eulerian Circuit

Ex (2):



A: Connected to B, C → **Degree = 2** (even)
B: Connected to A, C → **Degree = 2** (even)
C: Connected to A, B → **Degree = 2** (even)

✅ **Eulerian Circuit: A → B → C → A**
All vertices have even degree ✓
Uses every edge exactly once ✓
Returns to starting vertex ✓

Foundational Structures

C- Hamiltonian Circuit

Visits every nodes exactly once

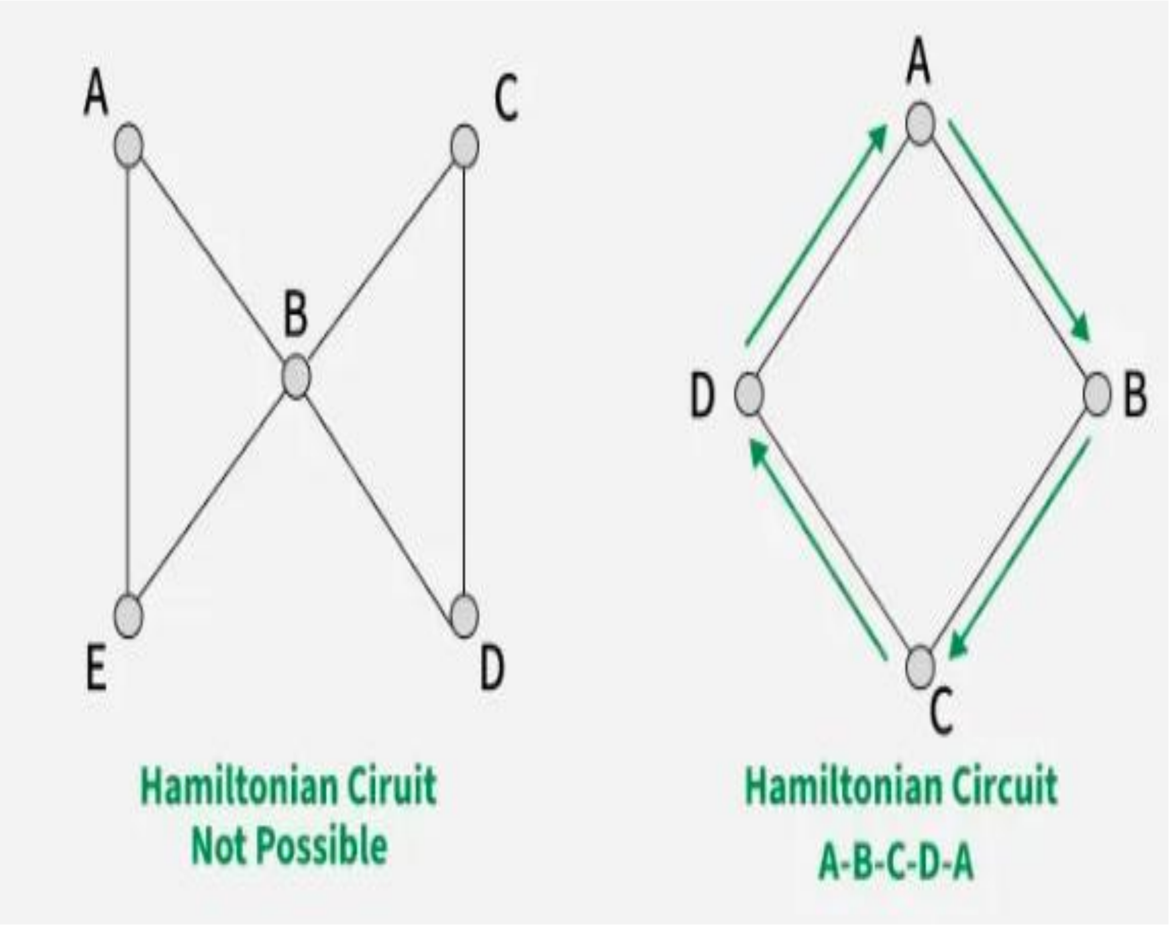
Hamiltonian Circuit Conditions:

- **Visits every node exactly once**
- **Returns to starting node**
- **No noderepetitions** (except start/end)
- Graph must have at least 3 node

Foundational Structures

C- Hamiltonian Circuit

Ex (1):



Foundational Structures

3. Connected Graph

A graph where there is a path between every pair of nodes.

Example (Connected):

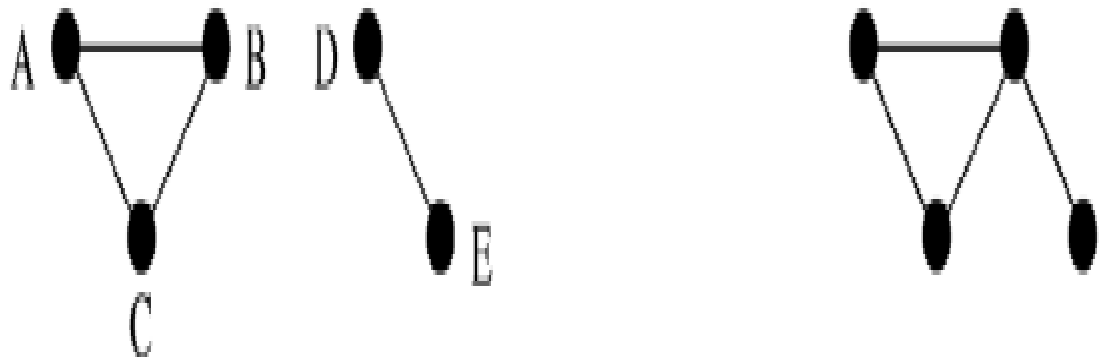
```
A --- B --- C
      |
      D
```

Example (Disconnected):

```
A --- B      C --- D
```

Foundational Structures

3. Connected Graph



(a) A nonconnected graph

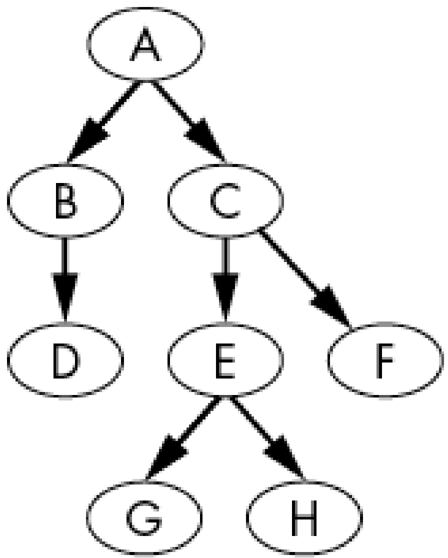
(b) A connected graph

Trees

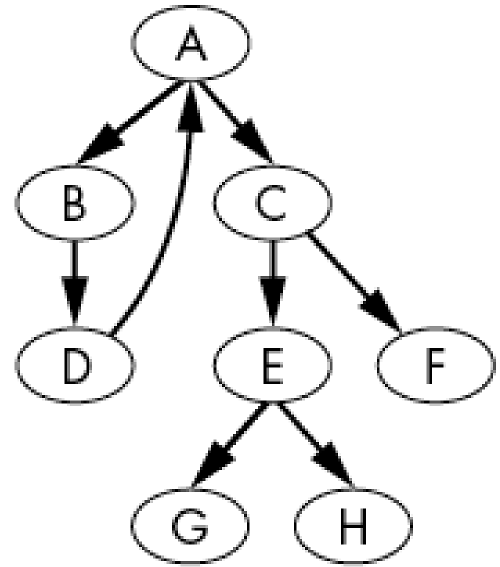
•**Trees: A Special Kind of Graph**.

•**Definition:** A tree is a connected, acyclic graph.

Acyclic: Contains no cycles (no loop you can traverse and return to the start).



Tree



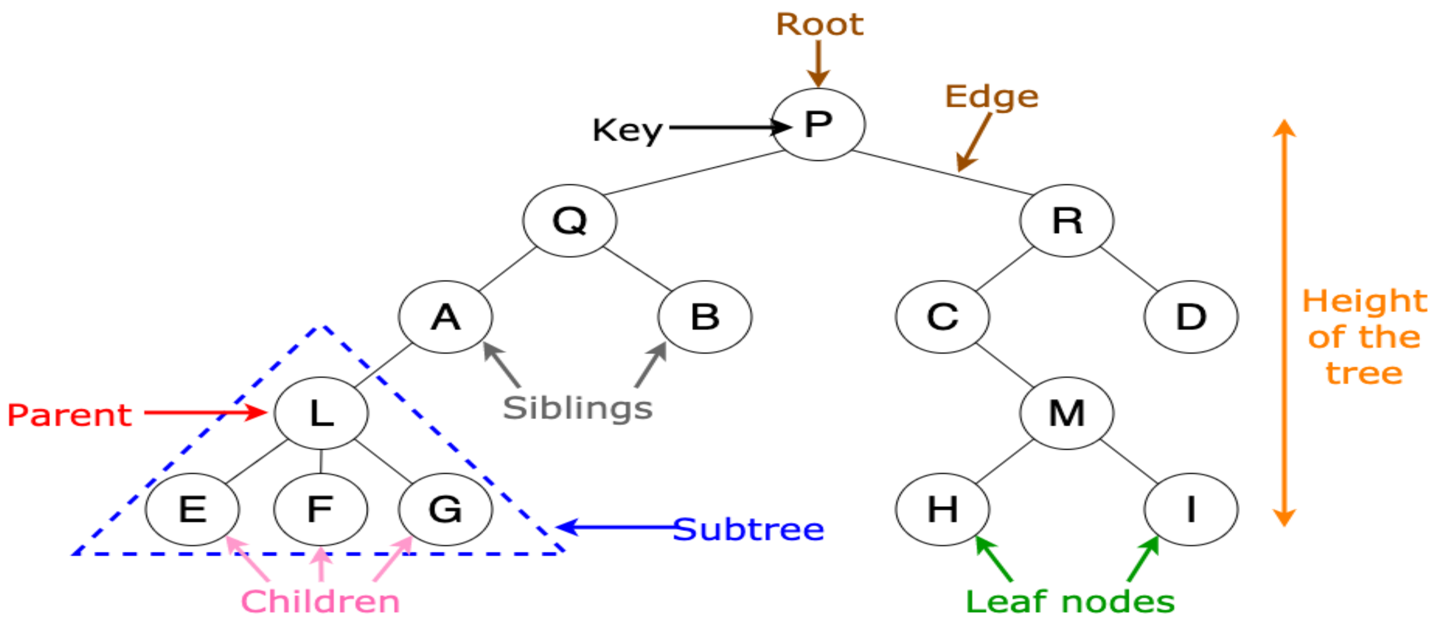
Not a tree
(child node loops back to an ancestor node)

Trees

Connected: There is a path between any two nodes.

Key Components:

Root Node: The topmost node.
Parent/Child Nodes: Hierarchical relationships.
Leaf Nodes: Nodes with no children.



Trees

Features

Every node, except the root, has exactly one parent.

Every node may give rise to zero or more child nodes.

A **binary tree** restricts the number of children per node to a maximum of two.

Degenerate trees have only a single pathway from root to its one leaf.

Trees

Example for Binary Tree