Artificial Intelligence

UNIT - IV: Expert system and applications

Syllabus:

Introduction phases in building expert systems, expert system versus traditional systems, rule-based expert systems, blackboard systems, model-based expert system, case-based expert system and hybrid expert system and application of expert systems.

Outcomes:

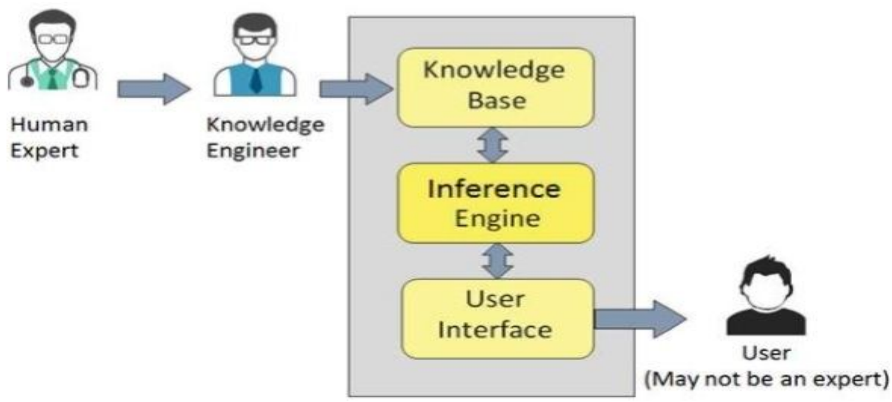
Student will be able to:

6.1 Introduction: Expert System

- The expert systems are the computer applications developed to solve complex problems in a particular domain, at the level of extra-ordinary human intelligence and expertise.
- **Characteristics of Expert Systems:**
    - High performance
    - Understandable
    - Reliable
    - Highly responsive

6.1.1 Components of Expert Systems

- The components of ES include −
    - Knowledge Base
    - Inference Engine
    - User Interface

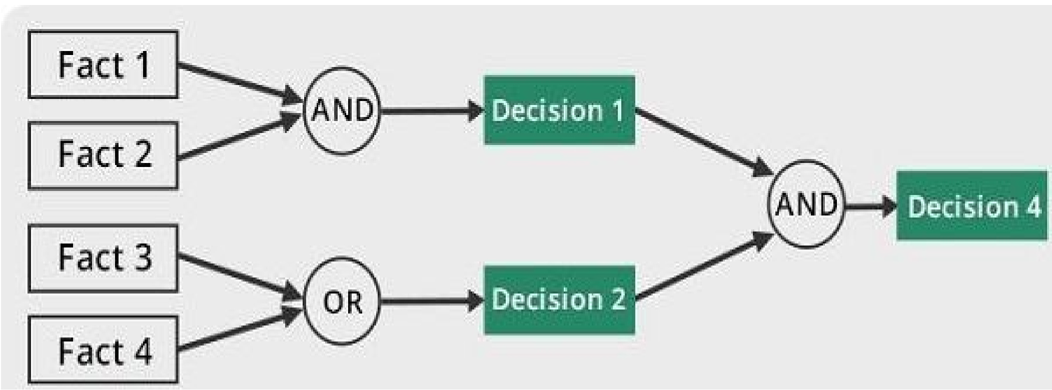


Knowledge Base:

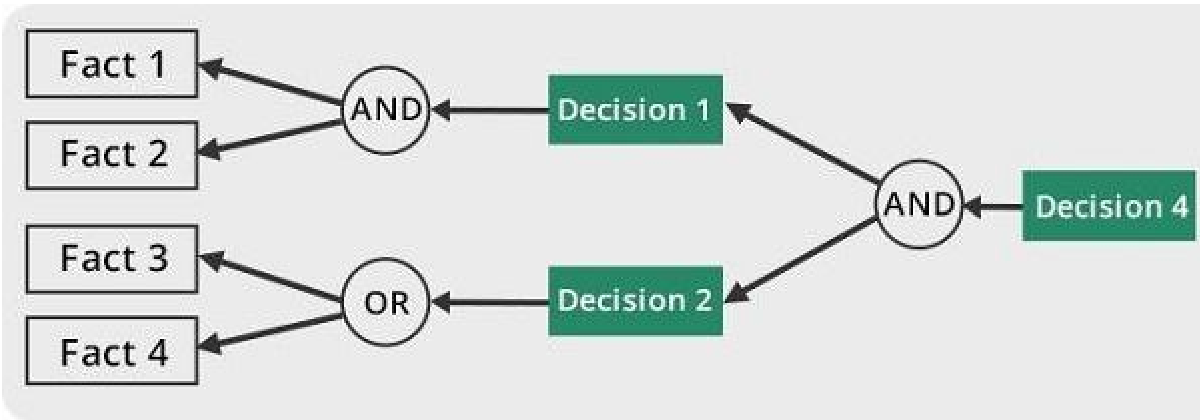
- It contains domain-specific and high-quality knowledge.
- Knowledge is required to exhibit intelligence. The success of any ES majorly depends upon the collection of highly accurate and precise knowledge.
- What is Knowledge?-The data is collection of facts. The information is organized as data and facts about the task domain. **Data, information,** and **past experience** combined together are termed as knowledge.
- **Components of Knowledge Base:** The knowledge base of an ES is a store of both, factual and heuristic knowledge.
    - **Factual Knowledge** − It is the information widely accepted by the Knowledge Engineers and scholars in the task domain.
    - **Heuristic Knowledge** − It is about practice, accurate judgement, one's ability of evaluation, and guessing.
- **Knowledge representation:** It is the method used to organize and formalize the knowledge in the knowledge base. It is in the form of IF-THEN-ELSE rules.
- **Knowledge Acquisition:** The success of any expert system majorly depends on the quality, completeness, and accuracy of the information stored in the knowledge base.
- The knowledge base is formed by readings from various experts, scholars, and the **Knowledge Engineers**. The knowledge engineer is a person with the qualities of empathy, quick learning, and case analyzing skills.
- He acquires information from subject expert by recording, interviewing, and observing him at work, etc. He then categorizes and organizes the information in a meaningful way, in the form of IF-THEN-ELSE rules, to be used by interference machine. The knowledge engineer also monitors the development of the ES.

Inference Engine:

- Use of efficient procedures and rules by the Inference Engine is essential in deducting a correct solution.
- In case of knowledge-based ES, the Inference Engine acquires and manipulates the knowledge from the knowledge base to arrive at a particular solution.
- In case of rule based ES, it −
  - Applies rules repeatedly to the facts, which are obtained from earlier rule application.
  - Adds new knowledge into the knowledge base if required.
  - Resolves rules conflict when multiple rules are applicable to a particular case.
- To recommend a solution, the Inference Engine uses the following strategies −
  - Forward Chaining
  - Backward Chaining
- **Forward Chaining:** It is a strategy of an expert system to answer the question, **"What can happen next?"**
  - Here, the Inference Engine follows the chain of conditions and derivations and finally deduces the outcome. It considers all the facts and rules, and sorts them before concluding to a solution.
  - This strategy is followed for working on conclusion, result, or effect. For example, prediction of share market status as an effect of changes in interest rates.



- **Backward Chaining:** With this strategy, an expert system finds out the answer to the question, **"Why this happened?"**
    - On the basis of what has already happened, the Inference Engine tries to find out which conditions could have happened in the past for this result. This strategy is followed for finding out cause or reason. For example, diagnosis of blood cancer in humans.



User Interface:

- User interface provides interaction between user of the ES and the ES itself. It is generally Natural Language Processing so as to be used by the user who is well-versed in the task domain. The user of the ES need not be necessarily an expert in Artificial Intelligence.
- It explains how the ES has arrived at a particular recommendation. The explanation may appear in the following forms −
    - Natural language displayed on screen.
    - Verbal narrations in natural language.
    - Listing of rule numbers displayed on the screen.
- The user interface makes it easy to trace the credibility of the deductions.
- **Requirements of Efficient ES User Interface:**
    - It should help users to accomplish their goals in shortest possible way.
    - It should be designed to work for user's existing or desired work practices.

- Its technology should be adaptable to user's requirements; not the other way round.
- It should make efficient use of user input.

6.2 Applications of Expert System

- The following table shows where ES can be applied.

Application	Description
Design Domain	Camera lens design, automobile design.
Medical Domain	Diagnosis Systems to deduce cause of disease from observed data, conduction medical operations on humans.
Monitoring Systems	Comparing data continuously with observed system or with prescribed behavior such as leakage monitoring in long petroleum pipeline.
Process Control Systems	Controlling a physical process based on monitoring.
Knowledge Domain	Finding out faults in vehicles, computers.
Finance/Commerce	Detection of possible fraud, suspicious transactions, stock market trading, Airline scheduling, cargo scheduling.

6.3 Phases in building Expert Systems

The process of ES development is iterative. Steps in developing the ES include −

- **Identify Problem Domain**
  - The problem must be suitable for an expert system to solve it.
  - Find the experts in task domain for the ES project.
  - Establish cost-effectiveness of the system.
- **Design the System**
  - Identify the ES Technology
  - Know and establish the degree of integration with the other systems and databases.
  - Realize how the concepts can represent the domain knowledge best.
- **Develop the Prototype**
  - From Knowledge Base: The knowledge engineer works to −
    - Acquire domain knowledge from the expert.
    - Represent it in the form of If-THEN-ELSE rules.
- **Test and Refine the Prototype**
  - The knowledge engineer uses sample cases to test the prototype for any deficiencies in performance.
  - End users test the prototypes of the ES.
- **Develop and Complete the ES**
  - Test and ensure the interaction of the ES with all elements of its environment, including end users, databases, and other information systems.
  - Document the ES project well.
  - Train the user to use ES.
- **Maintain the System**

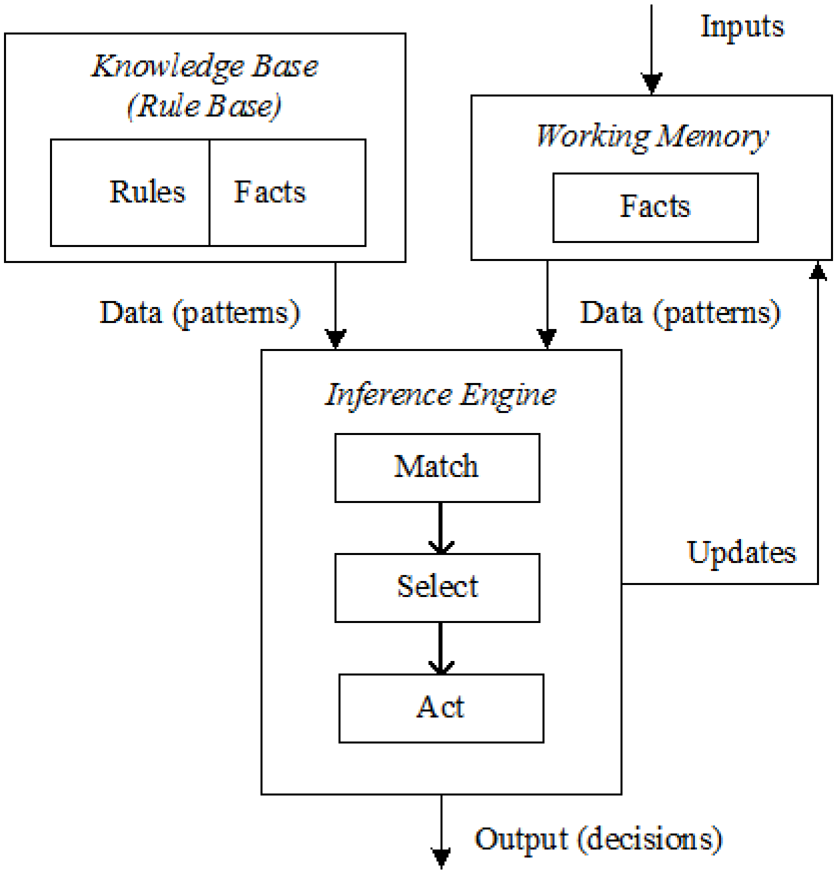
- Keep the knowledge base up-to-date by regular review and update.
- Cater for new interfaces with other information systems, as those systems evolve.

6.4 Expert system versus traditional systems

Expert System	Traditional System
The entire problem related expertise is encoded in data structures only, none is in programs.	Problem expertise is encoded in both program and data structures.
The use of knowledge is vital.	Data is used more efficiently than knowledge.
These are capable of explaining how a particular conclusion is reached and why requested information is needed during a process.	These are not capable of explaining a particular conclusion for a problem. These systems try to solve in a straight forward manner.
Problems are solved more efficiently	Not so efficient as an expert system
It uses the symbolic representations for knowledge i.e. the rules, different forms of networks, frames, scripts etc. and performs their inference through symbolic computations	These are unable to express in symbols. They just simplify the problems in a straight forward manner and are incapable to express the "how, why" questions.
Problem solving tools those are present in expert system	No problem solving tools in specific.
Solution of the problem is more accurate.	Solution of the problem may not be more accurate.
Provide a clear separation of knowledge from its processing.	Do not separate knowledge from the control structure to process this knowledge.
Process knowledge expressed in the form of rules and use symbolic reasoning to solve problems in a narrow domain.	Process data and use algorithms, a series of well-defined operations, to solve general numerical problems.
Trace the rules fired during a problem-solving session and explain how a particular conclusion was reached and why specific data was needed.	Do not explain how a particular result was obtained and why input data was needed.
Permit inexact reasoning and can deal with incomplete, uncertain and fuzzy data.	W ork only on problems where data is complete and exact.
Enhance the quality of problem solving by adding new rules or adjusting old ones in the knowledge base. W hen new knowledge is acquired, changes are easy to accomplish.	Enhance the quality of problem solving by changing the program code, which affects both the knowledge and its processing, making changes difficult.

6.5 Rule-based Systems

- **Rule-based systems** are used as a way to store and manipulate knowledge to interpret information in a useful way. They are often used in artificial intelligence applications and research.
- An RBS consists of a knowledge base and an inference engine. The knowledge base contains rules and facts.
- A typical rule-based system has four basic components:
  - **A list of rules or rule base**, which is a specific type of knowledge base.
  - **An inference engine or semantic reasoner**, which infers information or takes action based on the interaction of input and the rule base. The interpreter executes a production system program by performing the following **match-resolve-act cycle:**
    - **Match:** In this first phase, the left-hand sides of all productions are matched against the contents of working memory. As a result a conflict set is obtained, which consists of instantiations of all satisfied productions. An instantiation of a production is an ordered list of working memory elements that satisfies the left-hand side of the production.
    - **Conflict-Resolution:** In this second phase, one of the production instantiations in the conflict set is chosen for execution. If no productions are satisfied, the interpreter halts.
    - **Act:** In this third phase, the actions of the production selected in the conflict-resolution phase are executed. These actions may change the contents of working memory. At the end of this phase, execution returns to the first phase.
  - **Temporary working memory-** set of facts.
  - A **user interface** or other connection to the outside world through which input and output signals are received and sent.



- The most common RBS modes of operation are:
  - forward chaining (stimulus driven)
  - backward chaining (goal directed)
- **Forward Chaining:** Forward chaining mode of operation means that a rule is triggered when changes in the working memory produce a situation that matches all of its antecedents.

  Forward chaining is the process of inferring then-patterns from if-patterns that is consequents from antecedents. When an antecedent matches an assertion the antecedent is satisfied. When all antecedents of a rule are satisfied the rule is triggered. In deduction systems all triggered rules are allowed and may fire.

Forward Chaining Algorithm

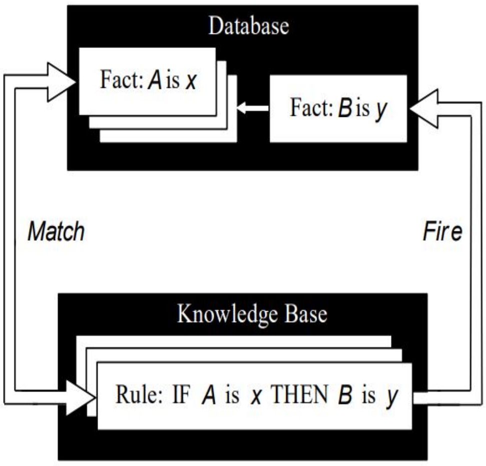
Repeat

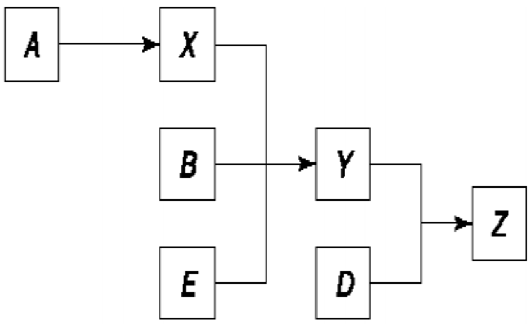
For each rule do

- **Match** all its antecedents to
  the facts from the Working memory
- if all antecedents of a rule are matched,
  Execute is consequents

until no rule produces a new assertion, or the goal is satisfied.

- **Backward Chaining:** The backward chaining mode of operation means that the systems begins with a goal and successively examines any rules with matching consequents. These candidate rules are considered one at a time. The unmet conditions are in turn reintroduced as new goals. The control procedure then shifts attention recursively toward the new goal. The effort terminates when the top goal is finally satisfied.
- **In a Rule-based System:**
  - The domain knowledge is represented by a set of IF-THEN production rules
  - Data is represented by a set of facts about the current situation.
  - The inference engine compares each rule stored in the knowledge base with facts contained in the database.
  - When the IF (condition) part of the rule matches a fact, the rule is fired and its THEN (action) part is executed.

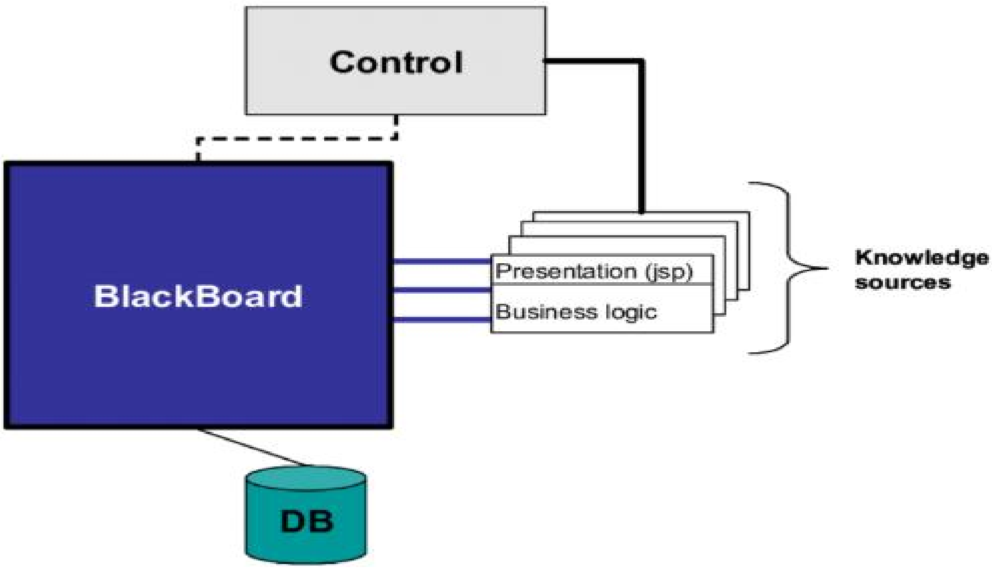


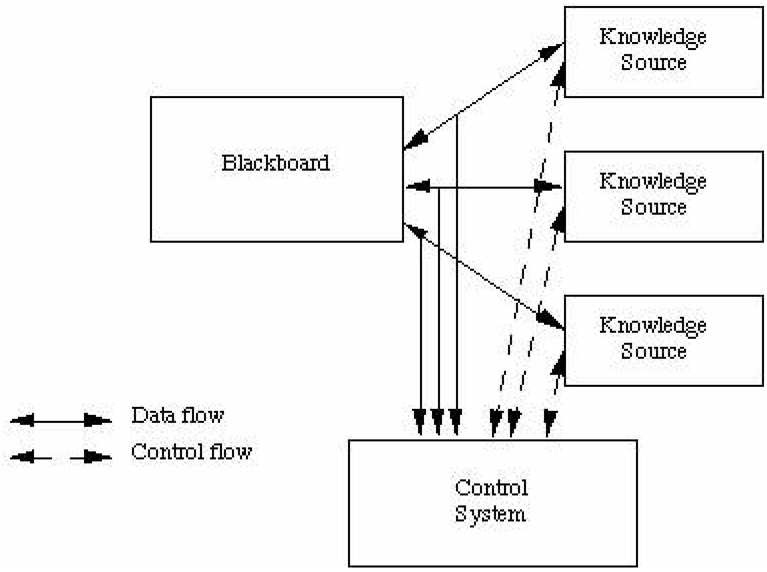


Rule 1: IF Y is true AND D is true THEN Z is true
Rule 2: IF X is true AND B is true AND E is true THEN Y is true
Rule 3: IF A is true THEN X is true

6.6 Blackboard System

- A blackboard system is an artificial intelligence approach based on the blackboard architectural model, where a common knowledge base, the "blackboard", is iteratively updated by a diverse group of specialist knowledge sources, starting with a problem specification and ending with a solution.
- Each knowledge source updates the blackboard with a partial solution when its internal constraints match the blackboard state. In this way, the specialists work together to solve the problem.
- The blackboard model was designed to handle complex, ill-defined problems, where the solution is the sum of its parts.
- A blackboard-system application consists of three major components:
  - The software specialist modules, which are called **knowledge sources (KSs)**. Like the human experts at a blackboard, each knowledge source provides specific expertise needed by the application.
  - **The blackboard**, a shared repository of problems, partial solutions, suggestions, and contributed information. The blackboard can be thought of as a dynamic "library" of contributions to the current problem that have been recently "published" by other knowledge sources.
  - **The control shell**, which controls the flow of problem-solving activity in the system. KSs need a mechanism to organize their use in the most effective and coherent fashion. In a blackboard system, this is provided by the control shell.

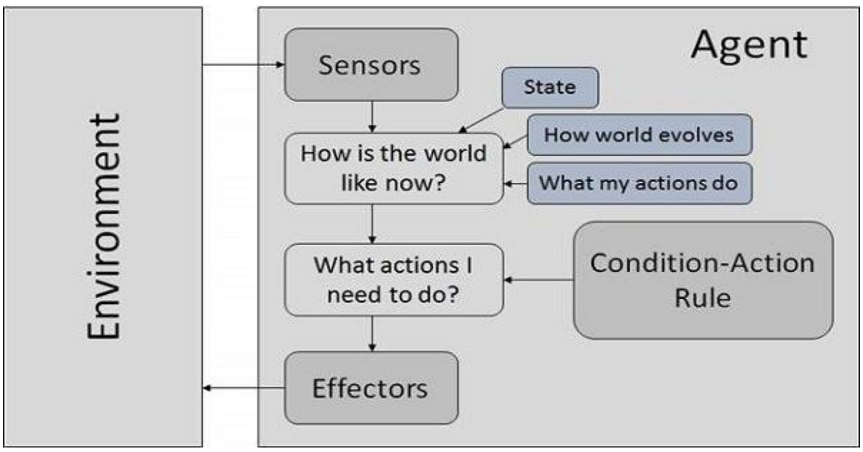




➢ The advantages of a blackboard include separation of knowledge into independent modules with each module being free to use the appropriate technology to arrive at the best solution with the most efficiency.

6.7 Model-based expert system

➢ They use a model of the world to choose their actions. They maintain an internal state.
- Model − knowledge about "how the things happen in the world".
- Internal State − It is a representation of unobserved aspects of current state depending on percept history.
- Updating the state requires the information about −
  - o How the world evolves.
  - o How the agent's actions affect the world.



- Model-based agent can handle a partially observable environment. Its current state is stored inside the agent maintaining some kind of structure which describes the part of the world which cannot be seen. This knowledge about "how the world works" is called a model of the world, hence the name "model-based agent".
- Model-based agents construct an internal representation of the world and use it to act.
- Model-based agents are made to deal with partial accessibility; they do this by keeping track of the part of the world it can see now. It does this by keeping an internal state that depends on what it has seen before so it holds information on the unobserved aspects of the current state.
- In a model-based reasoning system, knowledge can be represented using causal rules. For example, in a medical diagnosis system the knowledge base may contain the following rule:

  $\forall$patients : Stroke(patient) $\rightarrow$Confused(patient)

  Λ unequal(Pupils(patient))
- Models might be quantitative (for instance, based on mathematical equations) or qualitative (for instance, based on cause/effect models.) They may include representation of uncertainty. They might represent behaviour over time. They might represent "normal" behaviour, or might only represent abnormal behaviour.

6.8 Case-based expert system

- Case-based reasoning (CBR), broadly construed, is the process of solving new problems based on the solutions of similar past problems.
- Case-based reasoning has been formalized as a four-step process :
  - **Retrieve:** Given a target problem, retrieve from memory cases relevant to solving it. A case consists of a problem, its solution, and, typically, annotations about how the solution was derived. **Reuse:** Map the solution from the previous case to the target problem. This may involve adapting the solution as needed to fit the new situation.
  - **Revise:** Having mapped the previous solution to the target situation, test the new solution in the real world (or a simulation) and, if necessary, revise.
  - **Retain:** After the solution has been successfully adapted to the target problem, store the resulting experience as a new case in memory.
- An auto mechanic who fixes an engine by recalling another car that exhibited similar symptoms is using case-based reasoning. A lawyer who advocates a particular outcome in a trial based

on legal precedents or a judge who creates case law is using case-based reasoning.

- Case-based reasoning is a prominent type of analogy solution making.

6.9 Hybrid expert system

- Hybrid expert systems employ two or more representations of expertise to emulate reasoning in some domain.

Assignment-Cum-Tutorial Questions

Objective Questions

Section-A

1. Define the term "Expert System".
2. The components of expert system are ________________, _________________and ________________.
3. Knowledge comprises of ____________. [ ]

(a) Data (b) Information (c) Past Experience (d) All the above

4. The information that is widely accepted by the Knowledge Engineers and scholars in the task domain is called _________ knowledge. [ ]

(a) Factual (b) Heuristic (c) Domain (d) none

5. Knowledge that is about practice, accurate judgment, one's ability of evaluation, and guessing is called ______________ knowledge. [ ]

(a) Factual (b) Heuristic (c) Domain (d) none

6. ___________categorizes and organizes the information in a meaningful way. [ ]

(a) Knowledge Engineer (b) Human Expert (c) User (d) Tool

7. __________is a strategy of an expert system to answer the question, **"What can happen next?"** [ ]

(a) Forward Chaining (b) Backward chaining (c) both (d) none

8. __________is a strategy of an expert system finds out the answer to the question, **"Why this happened?"** [ ]

(a) Forward Chaining (b) Backward chaining (c) both (d) none

9. ___________ is an expert system without knowledge base. [ ]

(a) Tool (b) Shell (c) Task (d) none

10. In an Expert System, the entire problem related expertise is encoded in ______________________. [ ]

(i) Data Structures (ii) Programs

(a) Only (i) (b) Only (ii) (c) Both (i) and (ii) (d) none

11. In a traditional system, the entire problem related expertise is encoded in ______________________. [ ]

(i) Data Structures (ii) Programs

(a) Only (i) (b) Only (ii) (c) Both (i) and (ii) (d) none

12. The knowledgebase in a Rule-base system consists of ________.[ ]

(a) Rules (b) Facts (c) Both a & b (d) productions

13. Truth maintenance system supports ____________reasoning. [ ]

(a) Monotonic (b) Non-Monotonic (c) Both a & b (d) none

14. **MYCIN** is a ____________ expert system. [ ]

(a) Forward Chaining (b) Backward chaining (c) both (d) none

15. DENDRAL was written in the _____ programming language**.** **[ ]**

(a) PROLOG (b) LISP (c) FORTRAN (d) PYTHON

II) Descriptive Questions

1. List the characteristics and capabilities of Expert System.
2. Explain the components of an expert system.
3. Distinguish between Forward chaining and Backward chaining.
4. Enlist the application of Expert systems.
5. Describe the phases of developing an Expert system.
6. What do you mean by expert system technology? Explain.
7. Distinguish Expert system and Traditional system.
8. Explain about Rule-based Systems.
9. Explain Justification-based Truth maintenance system.
10. Write short notes on:

(i) MYCIN (ii) DENDRAl (iii) R1