



## **Expert Systems**

### **Artificial Intelligence**

*Expert system, topics - Introduction, expert system components and human interfaces, expert system characteristics, expert system features; Knowledge acquisition - issues and techniques; Knowledge base - representing and using domain knowledge, IF-THEN rules, semantic network, frames; Working memory; Inference engine : Forward chaining - data driven approach, Backward chaining - goal driven approach; Tree searches - DFS, BFS; Expert system shells - Shell components and description; Explanation - example, types of explanation; Application of expert systems.*

# Expert System

## Artificial Intelligence

### Topics

(Lectures 35, 36, 2 hours)

	Slides
<b>1. Introduction</b>	03-16
Expert system components and human interfaces, expert system characteristics, expert system features.	
<b>2. Knowledge Acquisition</b>	17-20
Issues and techniques.	
<b>3. Knowledge Base</b>	21-25
Representing and using domain knowledge - IF-THEN rules, semantic network, frames.	
<b>4. Working Memory</b>	26
<b>5. Inference Engine</b>	27-32
Forward chaining - data driven approach, backward chaining - goal driven approach, tree searches - DFS, BFS.	
<b>6. Expert System Shells</b>	33-34
Shell components and description.	
<b>7. Explanation</b>	35
Example, types of explanation	
<b>8. Application of Expert Systems</b>	36-37
<b>9. References</b>	38

## Expert System

### What is Expert System ?

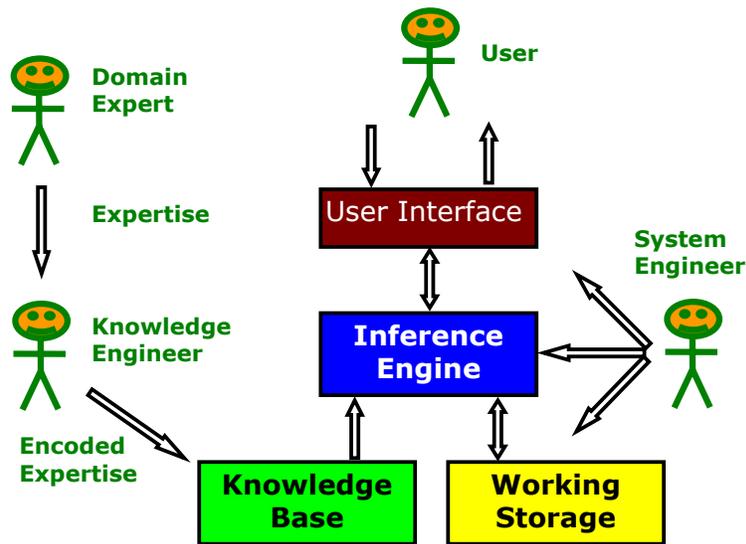
- An expert system, is an interactive computer-based decision tool that uses both facts and heuristics to solve difficult decision making problems, based on knowledge acquired from an expert.
- An expert system is a model and associated procedure that exhibits, within a specific domain, a degree of expertise in problem solving that is comparable to that of a human expert.
- An expert system compared with traditional computer :  
Inference engine + Knowledge = Expert system  
( Algorithm + Data structures = Program in traditional computer )
- First expert system, called DENDRAL, was developed in the early 70's at Stanford University.

## 1. Introduction

Expert systems are computer applications which embody some non-algorithmic expertise for solving certain types of problems. For example, expert systems are used in diagnostic applications. They also play chess, make financial planning decisions, configure computers, monitor real time systems, underwrite insurance policies, and perform many services which previously required human expertise.

### 1.1 Expert System Components And Human Interfaces

Expert systems have a number of major system components and interface with individuals who interact with the system in various roles. These are illustrated below.



Components of Expert System

The individual components and their roles are explained in next slides.

■ **Components and Interfaces**

- ‡ **Knowledge base** : A declarative representation of the expertise; often in IF THEN rules ;
- ‡ **Working storage** : The data which is specific to a problem being solved;
- ‡ **Inference engine** : The code at the core of the system which derives recommendations from the knowledge base and problem-specific data in working storage;
- ‡ **User interface** : The code that controls the dialog between the user and the system.

■ **Roles of Individuals who interact with the system**

- ‡ **Domain expert** : The individuals who currently are experts in solving the problems; here the system is intended to solve;
- ‡ **Knowledge engineer** : The individual who encodes the expert's knowledge in a declarative form that can be used by the expert system;
- ‡ **User** : The individual who will be consulting with the system to get advice which would have been provided by the expert.

## ■ Expert System Shells

Many expert systems are built with products called expert system shells. A shell is a piece of software which contains the user interface, a format for declarative knowledge in the knowledge base, and an inference engine. The knowledge and system engineers uses these shells in making expert systems.

‡ **Knowledge engineer** : uses the shell to build a system for a particular problem domain.

‡ **System engineer** : builds the user interface, designs the declarative format of the knowledge base, and implements the inference engine.

Depending on the size of the system, the knowledge engineer and the system engineer might be the same person.

## 2 Expert System Characteristics

Expert system operates as an interactive system that responds to questions, asks for clarifications, makes recommendations and generally aids the decision-making process.

Expert systems have many Characteristics :

### ■ Operates as an interactive system

This means an expert system :

- ‡ Responds to questions
- ‡ Asks for clarifications
- ‡ Makes recommendations
- ‡ Aids the decision-making process.

### ■ Tools have ability to sift (filter) knowledge

- ‡ Storage and retrieval of knowledge
- ‡ Mechanisms to expand and update knowledge base on a continuing basis.

### ■ Make logical inferences based on knowledge stored

- ‡ Simple reasoning mechanisms is used
- ‡ Knowledge base must have means of exploiting the knowledge stored, else it is useless; e.g., learning all the words in a language, without knowing how to combine those words to form a meaningful sentence.

■ **Ability to Explain Reasoning**

- ⌘ Remembers logical chain of reasoning; therefore user may ask
  - ◇ for explanation of a recommendation
  - ◇ factors considered in recommendation
- ⌘ Enhances user confidence in recommendation and acceptance of expert system

■ **Domain-Specific**

- ⌘ A particular system caters a narrow area of specialization; e.g., a medical expert system cannot be used to find faults in an electrical circuit.
- ⌘ Quality of advice offered by an expert system is dependent on the amount of knowledge stored.

■ **Capability to assign Confidence Values**

- ⌘ Can deliver quantitative information
- ⌘ Can interpret qualitatively derived values
- ⌘ Can address imprecise and incomplete data through assignment of confidence values.

■ **Applications**

- ‡ Best suited for those dealing with expert heuristics for solving problems.
- ‡ Not a suitable choice for those problems that can be solved using purely numerical techniques.

■ **Cost-Effective alternative to Human Expert**

- ‡ Expert systems have become increasingly popular because of their specialization, albeit in a narrow field.
- ‡ Encoding and storing the domain-specific knowledge is economic process due to small size.
- ‡ Specialists in many areas are rare and the cost of consulting them is high; an expert system of those areas can be useful and cost-effective alternative in the long run.

### 4.3 Expert System Features

The features which commonly exist in expert systems are :

- **Goal Driven Reasoning or Backward Chaining**

An inference technique which uses **IF-THEN rules** to repetitively break a goal into smaller sub-goals which are easier to prove;

- **Coping with Uncertainty**

The ability of the system to reason with rules and data which are not precisely known;

- **Data Driven Reasoning or Forward Chaining**

An inference technique which uses **IF-THEN rules** to deduce a problem solution from initial data;

- **Data Representation**

The way in which the problem specific data in the system is stored and accessed;

- **User Interface**

That portion of the code which creates an easy to use system;

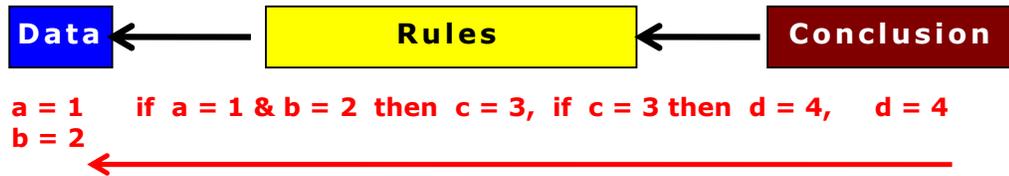
- **Explanations**

The ability of the system to explain the reasoning process that it used to reach a recommendation.

Each of these features were discussed in detail in previous lectures on AI. However for completion or easy to recall these are mentioned briefly here.

## Goal-Driven Reasoning

Goal-driven reasoning, or backward chaining, is an efficient way to solve problems. The algorithm proceeds from the desired goal, adding new assertions found.



The knowledge is structured in **rules** which describe how each of the possibilities might be selected.

The rule breaks the problem into sub-problems.

Example :

KB contains Rule set :

- Rule 1: If A and C                    Then F
- Rule 2: If A and E                    Then G
- Rule 3: If B                            Then E
- Rule 4: If G                             Then D

Problem : prove

    If A and B true                    Then D is true

## ● **Uncertainty**

Often the **Knowledge is imperfect** which causes uncertainty.

To work in the real world, Expert systems must be able to deal with uncertainty.

- one simple way is to associate a numeric value with each piece of information in the system.
- the numeric value represents the certainty with which the information is known.

There are different ways in which these numbers can be defined, and how they are combined during the inference process.

## Data Driven Reasoning

The data driven approach, or Forward chaining, uses rules similar to those used for backward chaining. However, the inference process is different. The system keeps track of the current state of problem solution and looks for rules which will move that state closer to a final solution. The Algorithm proceeds from a given situation to a desired goal, adding new assertions found.



**a = 1    if a = 1 & b = 2 then c = 3, if c = 3 then d = 4,    d = 4**  
**b = 2**



The knowledge is structured in **rules** which describe how each of the possibilities might be selected. The rule breaks the problem into sub-problems.

Example :

KB contains Rule set :

- |                    |        |
|--------------------|--------|
| Rule 1: If A and C | Then F |
| Rule 2: If A and E | Then G |
| Rule 3: If B       | Then E |
| Rule 4: If G       | Then D |

Problem : prove

    If A and B true                      Then D is true

## ● Data Representation

Expert system is built around a knowledge base module.

- knowledge acquisition is transferring knowledge from human expert to computer.
- Knowledge representation is faithful representation of what the expert knows.

No single knowledge representation system is optimal for all applications.

The success of expert system depends on choosing knowledge encoding scheme best for the kind of knowledge the system is based on.

The **IF-THEN rules**, **Semantic networks**, and **Frames** are the most commonly used representation schemes.

## ● **User Interface**

The acceptability of an expert system depends largely on the quality of the user interface.

- Scrolling dialog interface : It is easiest to implement and communicate with the user.
- Pop-up menus, windows, mice are more advanced interfaces and powerful tools for communicating with the user; they require graphics support.

## ● Explanations

An important features of expert systems is their ability to explain themselves.

Given that the system knows which rules were used during the inference process, the system can provide those rules to the user as means for explaining the results.

By looking at explanations, the knowledge engineer can see how the system is behaving, and how the rules and data are interacting.

This is very valuable diagnostic tool during development.

## 2. Knowledge Acquisition

Knowledge acquisition includes the elicitation, collection, analysis, modeling and validation of knowledge.

### 2.1 Issues in Knowledge Acquisition

The important issues in knowledge acquisition are:

- knowledge is in the head of experts
- Experts have vast amounts of knowledge
- Experts have a lot of tacit knowledge
  - ‡ They do not know all that they know and use
  - ‡ Tacit knowledge is hard (impossible) to describe
- Experts are very busy and valuable people
- One expert does not know everything
- Knowledge has a "shelf life"

## 2 Techniques for Knowledge Acquisition

The techniques for acquiring, analyzing and modeling knowledge are : Protocol-generation techniques, Protocol analysis techniques, Hierarchy-generation techniques, Matrix-based techniques, Sorting techniques, Limited-information and constrained-processing tasks, Diagram-based techniques. Each of these are briefly stated in next few slides.

- **Protocol-generation techniques**

Include many types of interviews (unstructured, semi-structured and structured), reporting and observational techniques.

- **Protocol analysis techniques**

Used with transcripts of interviews or text-based information to identify basic knowledge objects within a protocol, such as goals, decisions, relationships and attributes. These act as a bridge between the use of protocol-based techniques and knowledge modeling techniques.

- **Hierarchy-generation techniques**

Involve creation, reviewing and modification of hierarchical knowledge. Hierarchy-generation techniques, such as laddering, are used to build taxonomies or other hierarchical structures such as goal trees and decision networks. The Ladders are of various forms like concept ladder, attribute ladder, composition ladders.

- **Matrix-based techniques**

Involve the construction and filling-in a 2-D matrix (grid, table), indicating such things, as may be, for example, between concepts and properties (attributes and values) or between problems and solutions or between tasks and resources, etc. The elements within the matrix can contain: symbols (ticks, crosses, question marks ) , colors , numbers , text.

- **Sorting techniques**

Used for capturing the way people compare and order concepts; it may reveal knowledge about classes, properties and priorities.

- **Limited-information and constrained-processing tasks**

Techniques that either limit the time and/or information available to the expert when performing tasks. For example, a twenty-questions technique provides an efficient way of accessing the key information in a domain in a prioritized order.

- **Diagram-based techniques**

Include generation and use of concept maps, state transition networks, event diagrams and process maps. These are particularly important in capturing the "what, how, when, who and why" of tasks and events.

### 3. Knowledge Base (Representing and Using Domain Knowledge)

Expert system is built around a knowledge base module. Expert system contains a formal representation of the information provided by the domain expert. This information may be in the form of problem-solving rules, procedures, or data intrinsic to the domain. To incorporate these information into the system, it is necessary to make use of one or more knowledge representation methods. Some of these methods are described here.

Transferring knowledge from the human expert to a computer is often the most difficult part of building an expert system.

The knowledge acquired from the human expert must be encoded in such a way that it remains a faithful representation of what the expert knows, and it can be manipulated by a computer.

Three common methods of knowledge representation evolved over the years are **IF-THEN rules**, **Semantic networks** and **Frames**.

The first two methods were illustrated in the earlier lecture slides on knowledge representation therefore just mentioned here. The frame based representation is described more.

1 IF-THEN rules

Human experts usually tend to think along :

**condition**  $\Rightarrow$  **action**    or    **Situation**  $\Rightarrow$  **conclusion**

Rules **"if-then"** are predominant form of encoding knowledge in expert systems. These are of the form :

**If**             **$a_1, a_2, \dots, a_n$**

**Then**         **$b_1, b_2, \dots, b_n$**  where

each  **$a_i$**  is a condition or situation, and

each  **$b_i$**  is an action or a conclusion.

RC Chakraborty, www.readers.info

## Semantic Networks

In this scheme, knowledge is represented in terms of objects and relationships between objects.

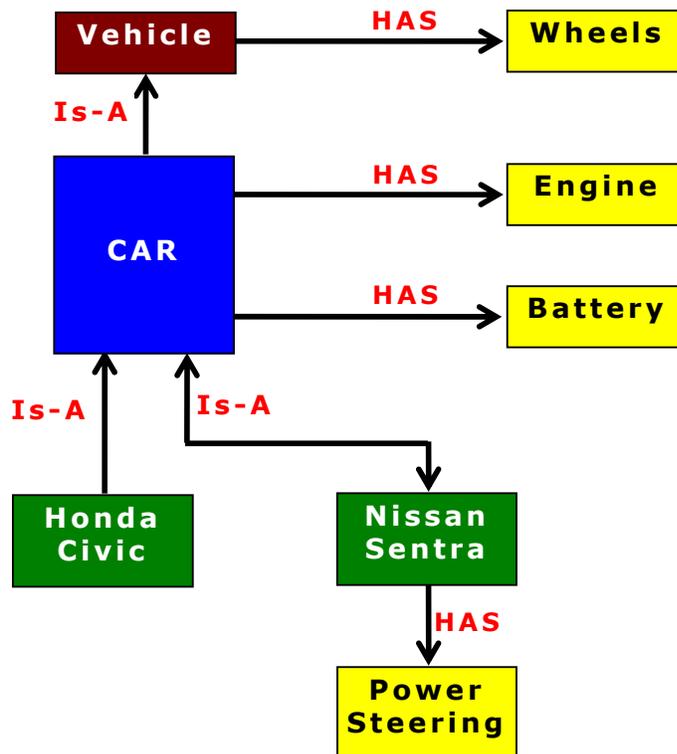
The objects are denoted as nodes of a graph. The relationship between two objects are denoted as a link between the corresponding two nodes.

The most common form of semantic networks uses the links between nodes to represent **IS-A** and **HAS** relationships between objects.

### Example of Semantic Network

The Fig. below shows a car **IS-A** vehicle; a vehicle **HAS** wheels.

This kind of relationship establishes an inheritance hierarchy in the network, with the objects lower down in the network inheriting properties from the objects higher up.



### **Frames**

In this technique, knowledge is decomposed into highly modular pieces called frames, which are generalized record structures. Knowledge consist of concepts, situations, attributes of concepts, relationships between concepts, and procedures to handle relationships as well as attribute values.

- ‡ Each concept may be represented as a separate **frame**.
- ‡ The attributes, the relationships between concepts, and the procedures are allotted to **slots** in a frame.
- ‡ The contents of a slot may be of any **data type** - numbers, strings, functions or procedures and so on.
- ‡ The frames may be linked to other frames, providing the same kind of inheritance as that provided by a semantic network.

A frame-based representation is ideally suited for objected-oriented programming techniques. An example of Frame-based representation of knowledge is shown in next slide.

**Example : Frame-based Representation of Knowledge.**

Two frames, their slots and the slots filled with data type are shown.

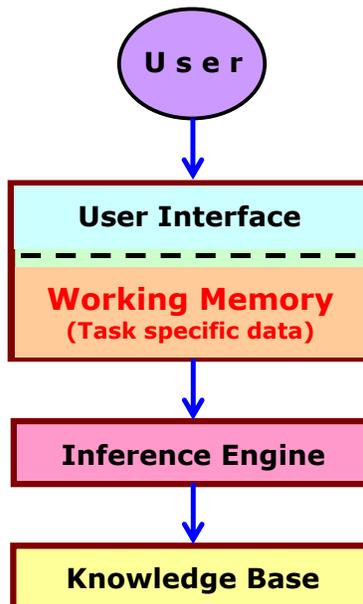
<b>Frame</b>	<i>Car</i>
<b>Inheritance Slot</b>	<i>Is-A</i>
<b>Value</b>	<i>Vehicle</i>
<b>Attribute Slot</b>	<i>Engine</i>
<b>Value</b>	<i>Vehicle</i>
<b>Value</b>	<i>1</i>
<b>Value</b>	
<b>Attribute Slot</b>	<i>Cylinders</i>
<b>Value</b>	<i>4</i>
<b>Value</b>	<i>6</i>
<b>Value</b>	<i>8</i>
<b>Attribute Slot</b>	<i>Doors</i>
<b>Value</b>	<i>2</i>
<b>Value</b>	<i>5</i>
<b>Value</b>	<i>4</i>

<b>Frame</b>	<i>Car</i>
<b>Inheritance Slot</b>	<i>Is-A</i>
<b>Value</b>	<i>Car</i>
<b>Attribute Slot</b>	<i>Make</i>
<b>Value</b>	<i>Honda</i>
<b>Value</b>	
<b>Value</b>	
<b>Attribute Slot</b>	<i>Year</i>
<b>Value</b>	<i>1989</i>
<b>Value</b>	
<b>Value</b>	
<b>Attribute Slot</b>	
<b>Value</b>	
<b>Value</b>	
<b>Value</b>	

#### 4. Working Memory

Working memory refers to task-specific data for a problem. The contents of the working memory, changes with each problem situation. Consequently, it is the most dynamic component of an expert system, assuming that it is kept current.

- ‡ Every problem in a domain has some unique data associated with it.
- ‡ Data may consist of the set of conditions leading to the problem, its parameters and so on.
- ‡ Data specific to the problem needs to be input by the user at the time of using, means consulting the expert system. The Working memory is related to user interface
- ‡ Fig. below shows how Working memory is closely related to user interface of the expert system.



## 5. Inference Engine

The inference engine is a generic control mechanism for navigating through and manipulating knowledge and deduce results in an organized manner. The inference engine's generic control mechanism applies the axiomatic (self-evident) knowledge present in the knowledge base to the task-specific data to arrive at some conclusion.

- ‡ Inference engine the other key component of all expert systems.
- ‡ Just a knowledge base alone is not of much use if there are no facilities for navigating through and manipulating the knowledge to deduce something from knowledge base.
- ‡ A knowledge base is usually very large, it is necessary to have inferencing mechanisms that search through the database and deduce results in an organized manner.

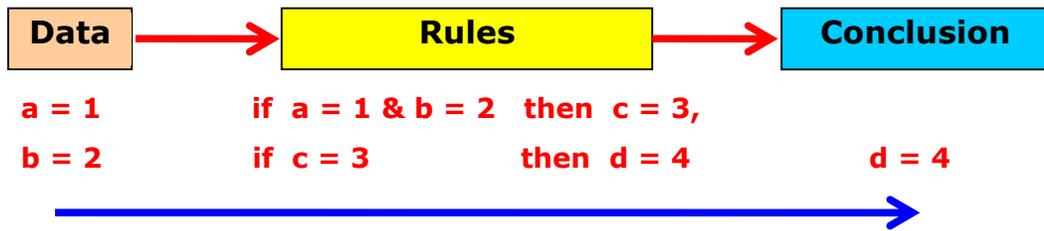
The Forward chaining, Backward chaining and Tree searches are some of the techniques used for drawing inferences from the knowledge base.

*These techniques were talked in the earlier lectures on Problem Solving : Search and Control Strategies, and Knowledge Representation. However they are relooked in the context of expert system.*

### 5.1 Forward Chaining Algorithm

Forward chaining is a techniques for drawing inferences from Rule base. Forward-chaining inference is often called **data driven**.

- ⌘ The algorithm proceeds from a given situation to a desired goal, adding new assertions (facts) found.
- ⌘ A forward-chaining, system compares data in the working memory against the conditions in the **IF** parts of the **rules** and determines which rule to fire.
- ⌘ Data Driven



⌘ Example : Forward Channing

■ Given : A Rule base contains following Rule set

- Rule 1: If A and C                      Then F
- Rule 2: If A and E                      Then G
- Rule 3: If B                              Then E
- Rule 4: If G                              Then D

■ Problem : Prove

    If A and B true                      Then D is true

*[Continued in next slide]*

[Continued from previous slide]

■ **Solution :**

- (i) † Start with input given **A, B** is true and then
- † start at **Rule 1** and go forward/down till a rule "fires" is found.

First iteration :

- (ii) † **Rule 3** fires : conclusion **E** is true
- † new knowledge found
- (iii) † No other rule fires;
- † end of first iteration.
- (iv) † Goal not found;
- † new knowledge found at (ii);
- † go for second iteration

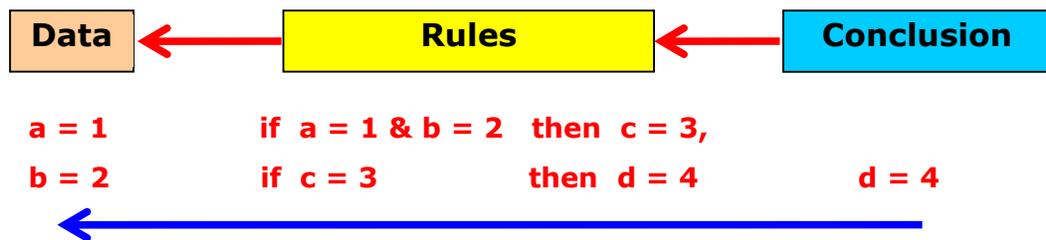
Second iteration :

- (v) † **Rule 2** fires : conclusion **G** is true
- † new knowledge found
- (vi) † **Rule 4** fires : conclusion **D** is true
- † Goal found;
- † Proved

## 5.2 Backward Chaining Algorithm

Backward chaining is a techniques for drawing inferences from Rule base. Backward-chaining inference is often called **goal driven**.

- ‡ The algorithm proceeds from desired goal, adding new assertions found.
- ‡ A backward-chaining, system looks for the action in the THEN clause of the rules that matches the specified goal.
- ‡ Goal Driven



‡ Example : Backward Channing

■ Given : Rule base contains following Rule set

- |                    |        |
|--------------------|--------|
| Rule 1: If A and C | Then F |
| Rule 2: If A and E | Then G |
| Rule 3: If B       | Then E |
| Rule 4: If G       | Then D |

■ Problem : Prove

If A and B true                    Then D is true

*[Continued in next slide]*

[Continued from previous slide]

■ **Solution :**

- (i) † Start with goal ie **D** is true
- † go backward/up till a rule "fires" is found.

First iteration :

- (ii) † **Rule 4** fires :
- † new sub goal to prove **G** is true
- † go backward
- (iii) † **Rule 2** "fires"; conclusion: **A** is true
- † new sub goal to prove **E** is true
- † go backward;
- (iv) † no other rule fires; end of first iteration.
- † new sub goal found at (iii);
- † go for second iteration

Second iteration :

- (v) † **Rule 3** fires :
- † conclusion **B** is true (2nd input found)
- † both inputs **A** and **B** ascertained
- † Proved

### 5.3 Tree Searches

Often a knowledge base is represented as a branching network or tree. Many tree searching algorithms exist but two basic approaches are depth-first search and breadth-first search.

*Note : Here these two search are briefly mentioned since they were described with examples in the previous lectures.*

#### ■ Depth-First Search

- ‡ Algorithm begins at initial node
- ‡ Check to see if the left-most below initial node (call node **A**) is a goal node.
- ‡ If not, include node **A** on a list of sub-goals outstanding.
- ‡ Then starts with node **A** and looks at the first node below it, and so on.
- ‡ If no more lower level nodes, and goal node not reached, then start from last node on outstanding list and follow next route of descent to the right.

#### ■ Breadth-First Search

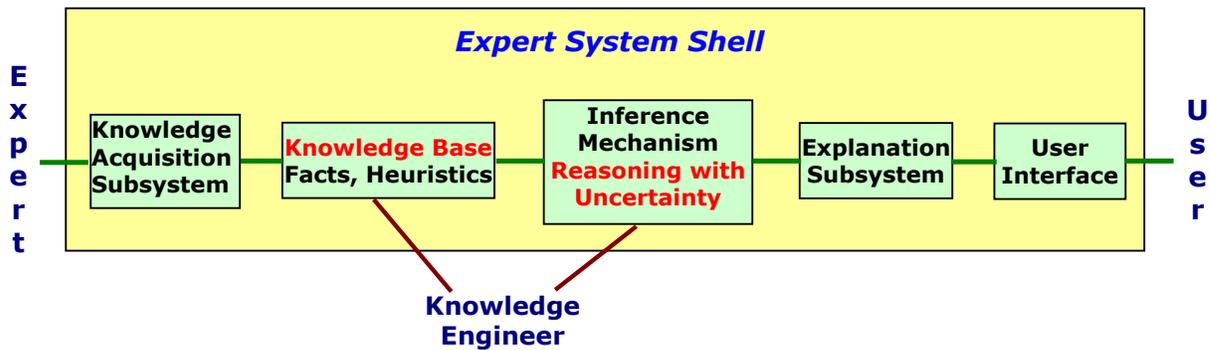
- ‡ Algorithm starts by expanding all the nodes one level below the initial node.
- ‡ Expand all nodes till a solution is reached or the tree is completely expanded.
- ‡ Find the shortest path from initial assertion to a solution.

## 6. Expert System Shells

An Expert system shell is a software development environment. It contains the basic components of expert systems. A shell is associated with a prescribed method for building applications by configuring and instantiating these components.

### 6.1 Shell components and description

The generic components of a shell : the knowledge acquisition, the knowledge Base, the reasoning, the explanation and the user interface are shown below. The knowledge base and reasoning engine are the core components.



All these components are described in the next slide.

[continuing from previous slide]

- **Knowledge Base**

A store of factual and heuristic knowledge. Expert system tool provides one or more knowledge representation schemes for expressing knowledge about the application domain. Some tools use both **Frames** (objects) and **IF-THEN rules**. In **PROLOG** the knowledge is represented as logical statements.

- **Reasoning Engine**

Inference mechanisms for manipulating the symbolic information and knowledge in the knowledge base form a line of reasoning in solving a problem. The inference mechanism can range from simple modus ponens backward chaining of **IF-THEN rules** to **Case-Based reasoning**.

- **Knowledge Acquisition subsystem**

A subsystem to help experts in build knowledge bases. However, collecting knowledge, needed to solve problems and build the knowledge base, is the biggest bottleneck in building expert systems.

- **Explanation subsystem**

A subsystem that explains the system's actions. The explanation can range from how the final or intermediate solutions were arrived at justifying the need for additional data.

- **User Interface**

A means of communication with the user. The user interface is generally not a part of the expert system technology. It was not given much attention in the past. However, the user interface can make a critical difference in the perceived utility of an Expert system.

## 7. Explanation

Most expert systems have explanation facilities that allow the user to ask questions - **why and how it reached some conclusion.**

The questions are answered by referring to the system goals, the rules being used, and existing problem solving. The rules typically reflect empirical, or "compiled" knowledge. They are codes of an expert's rules of thumb, not the expert's deeper understanding.

### Example :

Dialog with an expert system designed to give advice on car problems.

System Does the car start ?

User No.

System Does the engine turn over ?

User Yes.

System Do you smell gas ?

User Yes

System Recommendation - Wait 5 minutes and try again.

User Why ?

System I used the rule:

If not start, and engine\_turn\_over, and smell\_gas

Then recommendation is 'Wait 5 minutes and try again'

User

Note : The rule gives the correct advice for a flooded car, and knows the questions to be ask to determine if the car is flooded, but it does not contain the knowledge of **what** a flooded car is and **why** waiting will help.

### Types of Explanation

There are four types of explanations commonly used in expert systems.

- ‡ Rule trace reports on the progress of a consultation;
- ‡ Explanation of **how** the system reached to the given conclusion;
- ‡ Explanation of **why** the system did not give any conclusion.
- ‡ Explanation of **why** the system is asking a question;

## 8. Application of Expert Systems

The Expert systems have found their way into most areas of knowledge work. The applications of expert systems technology have widely proliferated to industrial and commercial problems, and even helping NASA to plan the maintenance of a space shuttle for its next flight. The main applications are stated in next few slides.

### ‡ **Diagnosis and Troubleshooting of Devices and Systems**

Medical diagnosis was one of the first knowledge areas to which Expert system technology was applied in 1976. However, the diagnosis of engineering systems quickly surpassed medical diagnosis.

### ‡ **Planning and Scheduling**

The Expert system's commercial potential in planning and scheduling has been recognized as very large. Examples are airlines scheduling their flights, personnel, and gates; the manufacturing process planning and job scheduling;

### ‡ **Configuration of Manufactured Objects from sub-assemblies**

Configuration problems are synthesized from a given set of elements related by a set of constraints. The Expert systems have been very useful to find solutions. For example, modular home building and manufacturing involving complex engineering design.

### **‡ Financial Decision Making**

The financial services are the vigorous user of expert system techniques. Advisory programs have been created to assist bankers in determining whether to make loans to businesses and individuals. Insurance companies to assess the risk presented by the customer and to determine a price for the insurance. ES are used in typical applications in the financial markets / foreign exchange trading.

### **‡ Knowledge Publishing**

This is relatively new, but also potentially explosive area. Here the primary function of the Expert system is to deliver knowledge that is relevant to the user's problem. The two most widely known Expert systems are : one, an **advisor** on appropriate grammatical usage in a text; and the other, is a **tax advisor** on tax strategy, tactics, and individual tax policy.

### **‡ Process Monitoring and Control**

Here Expert system does analysis of real-time data from physical devices, looking for anomalies, predicting trends, controlling optimality and failure correction. Examples of real-time systems that actively monitor processes are found in the steel making and oil refining industries.

### **‡ Design and Manufacturing**

Here the Expert systems assist in the design of physical devices and processes, ranging from high-level conceptual design of abstract entities all the way to factory floor configuration of manufacturing processes.

**9. References : Textbooks**

1. "Artificial Intelligence", by Elaine Rich and Kevin Knight, (2006), McGraw Hill companies Inc., Chapter 20, page 547-557.
2. "Introduction To Artificial Intelligence & Expert Systems", by Dan W Patterson, (2007), Prentice-hall, page 1-464.
3. "Expert Systems: Introduction To First And Second Generation And Hybrid Knowledge Based Systems", by Chris Nikolopoulos, (1997), Merzell Dekker INC, page 1-325
4. "Artificial intelligence and expert systems for engineers", by C. S. Krishnamoorthy, S. Rajeev, (1996), CRC Press INC, page 1-293.
5. Related documents from open source, mainly internet. An exhaustive list is being prepared for inclusion at a later date.