Expert System

What is Expert System ?

- An expert system is a computer program that simulates the thought process of a human expert to solve complex decision problems in a specific domain.

• 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)

The Need for Expert Systems

Expert systems are necessitated by the limitations associated with conventional human decision-making processes, including:
1. Human expertise is very scarce.
2. Humans get tired from physical or mental workload.
3. Humans forget crucial details of a problem.
4. Humans are inconsistent in their day-to-day decisions.
5. Humans have limited working memory.
6. Humans are unable to comprehend large amounts of data quickly.
7. Humans are unable to retain large amounts of data in memory.
8. Humans are slow in recalling information stored in memory.
9. Humans are subject to deliberate or inadvertent bias in their actions.
10. Humans can deliberately avoid decision responsibilities.
11. Humans *lie*, *hide*, and *die.*

Coupled with these human limitations are the weaknesses inherent in conventional programming and traditional decision-support tools.
Conventional programs:

1. Are algorithmic in nature and depend only on raw machine power
2. Depend on facts that may be difficult to obtain
3. Do not make use of the effective heuristic approaches used by human experts
4. Are not easily adaptable to changing problem environments
5. Seek explicit and factual solutions that may not be possible

Benefits of Expert Systems

Expert systems offer an environment where the good capabilities of humans and the power of computers can be incorporated to overcome many of the limitations discussed in the previous section.
Expert systems:
1. Increase the probability, frequency, and consistency of making good decisions
2. Help distribute human expertise
3. Facilitate real-time, low-cost expert-level decisions
4. Enhance the utilization of most of the available data
5. Permit objectivity by weighing evidence without bias and without regard for the user's personal and emotional reaction
6. Permit dynamism through modularity of structure
7. Free up the mind and time of the human expert to enable him or her to concentrate on more creative activities
8. Encourage investigations into the subtle areas of a problem

Applications of Expert System

Table 1. Expert/Knowledge-Based System Application Areas

Problem Type	Description
Control	Governing system behavior to meet specifications
Design	Configuring Objects under constraint
Diagnosis	Inferring System Malfunction from observables
Instruction	Diagnosing, debugging, and repairing student behavior
Interpretation	Inferring situation description from data
Monitoring	Comparing observations to expectations
Planning	Designing actions
Prediction	Inferring likely consequences of given situation
Prescription	Recommending solution to system malfunction
Selection	Identifying best choice from a list of possibilities

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

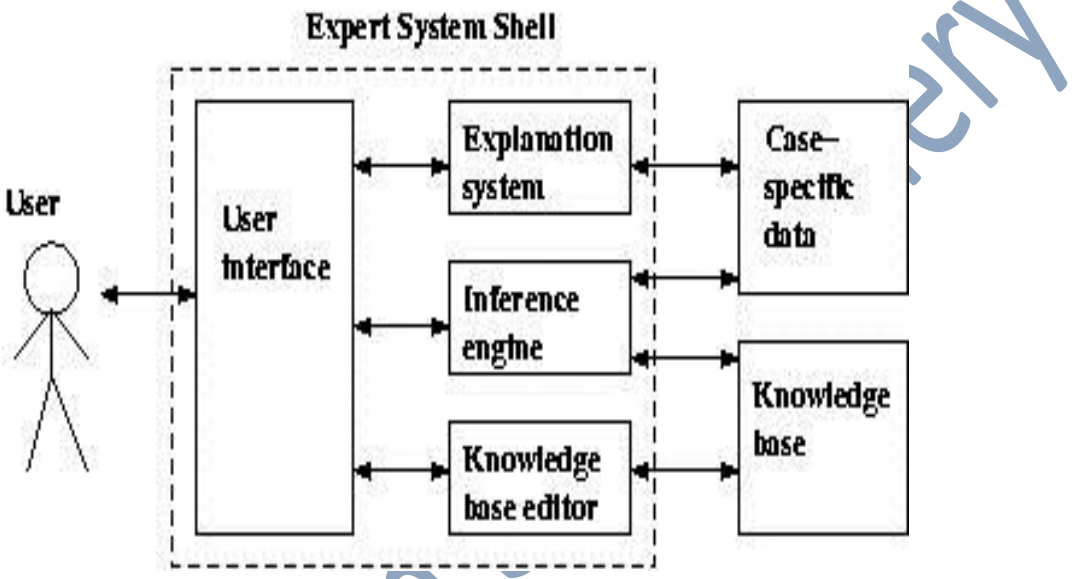
‡ Best suited for those dealing with expert heuristics for solving problems.
‡ Not a suitable choice for those problems that can be solved using purely numerical techniques.

■ **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.

Expert System Architecture



- Figure shows the most important modules that make up a rule-based expert system.
- The user interacts with the system through a
- *user interface* which may use menus, natural language or any other style of interaction).
- The core components of expert systems are the knowledge base and the reasoning engine.
- Then an *inference engine* is used to reason with both the *expert knowledge*) extracted from our friendly expert) and data specific to the particular problem being solved. The expert knowledge will typically be in the form of a set of IF-THEN rules.
- The *case specific data* includes both data provided by the user and partial conclusions (along with certainty measures) based on this data.

- Almost all expert systems also have an *explanation subsystem* ، which allows the program to explain its reasoning to the user. Some systems also have a
- *knowledge base editor* which help the expert or knowledge engineer to easily update and check the knowledge base .
- One important feature of expert systems is the way they (usually) separate domain specific knowledge from more general purpose reasoning and representation techniques.
- The general purpose bit (in the dotted box in the figure) is referred to as an *expert system shell* .As we see in the figure, the shell will provide the inference engine (and knowledge representation scheme), a user interface, an explanation system and sometimes a knowledge base editor.
- Using shells to write expert systems generally greatly reduces the cost and time of development.

Data, Information and Knowledge

- **Data** — measurements or records about events (prices, temperature, etc). Data can be numerical, alphabetical, images, sounds, etc.
- **Information** — analyzed and organized data such that we know its characteristics (average, range, variance, distributions, clusters, etc).
- **Knowledge** — information put into a specific context (e.g. distribution of oil prices, a map of London, etc).).**Knowledge** is a theoretical or practical understanding of a subject or a domain. Those who possess knowledge are called experts.
- **Knowledge** is also the sum of what is currently known. Those who possess knowledge are called experts.
- Anyone can be considered a **domain expert** if he or she has deep knowledge (of both facts and rules) and strong practical experience in a particular domain. The area of the domain may be limited. In general, an expert is a skilful person who can do things other people cannot.

Knowledge Acquisition

- knowledge acquisition is transferring knowledge from human expert to computer. Knowledge acquisition includes the elicitation, collection, analysis, modeling and validation of knowledge.

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 Representation

- Expert system is built around a knowledge base module.
- 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.
- The human mental process is internal, and it is too complex to be represented as an algorithm. However, most experts are capable of expressing their knowledge in the form of **rules** for problem solving.

IF the 'traffic light' is green
THEN the action is go

IF the 'traffic light' is red
THEN the action is stop

Rules as a knowledge representation technique

The term ***rule*** in AI, which is the most commonly used type of knowledge representation, can be defined as an IF-THEN structure that relates given information or facts in the IF part to some action in the THEN part. A rule provides some description of how to solve a problem. Rules are relatively easy to create and understand.

Any rule consists of two parts: the IF part, called the ***antecedent*** (***premise*** or ***condition***) and the THEN part called the ***consequent*** (***conclusion*** or ***action***).

IF <antecedent>

THEN <consequent>

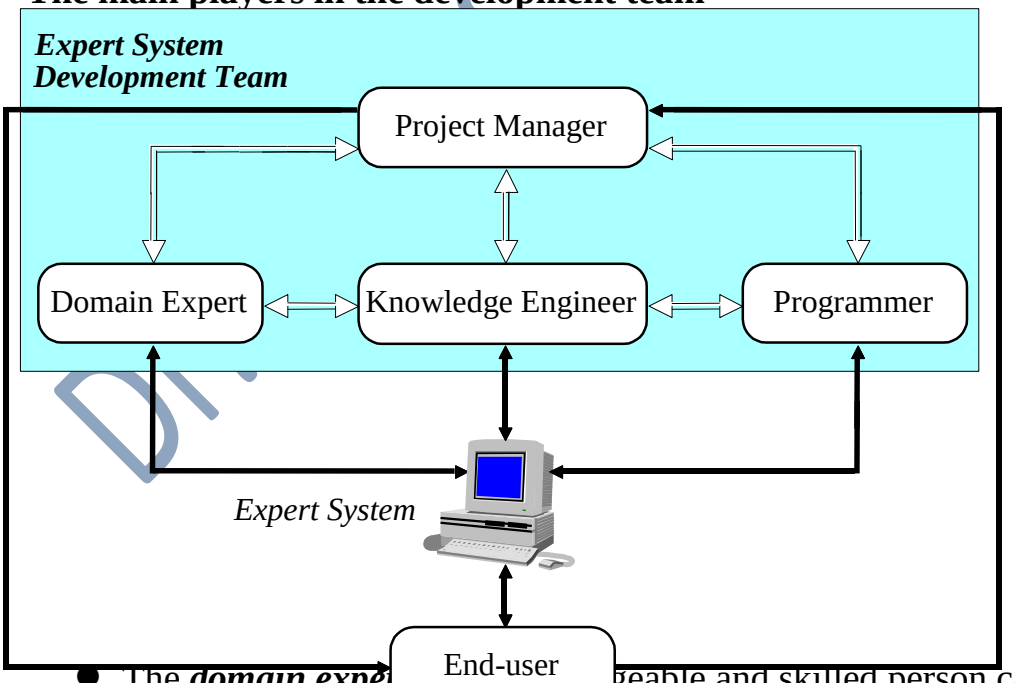
A rule can have multiple antecedents joined by the keywords **AND** (**conjunction**), **OR** (**disjunction**) or a combination of both.

IF <antecedent 1>	IF <antecedent 1>
AND <antecedent 2>	OR <antecedent 2>
.	.
.	.
.	.
AND <antecedent *n*>	OR <antecedent *n*>
THEN <consequent>	THEN <consequent>

The main players in the development team

- There are five members of the expert system development team: the **domain expert**, the **knowledge engineer**, the **programmer**, the **project manager** and the **end-user**.
- The success of their expert system entirely depends on how well the members work together.

The main players in the development team



- The ***domain expe*** geable and skilled person capable of solving problems in a specific area or ***domain***. This person has the greatest expertise in a given domain. This expertise is to be captured in the expert system. Therefore, the expert must be able to communicate his or her knowledge, be willing to participate in

the expert system development and commit a substantial amount of time to the project. The domain expert is the most important player in the expert system development team.

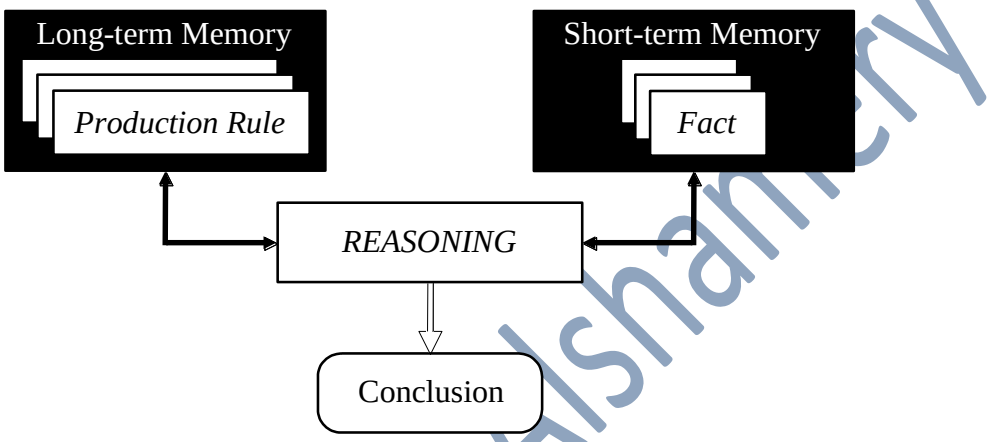
- The ***knowledge engineer*** is someone who is capable of designing, building and testing an expert system. He or she interviews the domain expert to find out how a particular problem is solved. The knowledge engineer establishes what reasoning methods the expert uses to handle facts and rules and decides how to represent them in the expert system. The knowledge engineer then chooses some development software or an expert system shell, or looks at programming languages for encoding the knowledge. And finally, the knowledge engineer is responsible for testing, revising and integrating the expert system into the workplace.
- The ***programmer*** is the person responsible for the actual programming, describing the domain knowledge in terms that a computer can understand. The programmer needs to have skills in symbolic programming in such AI languages as LISP, Prolog and also some experience in the application of different types of expert system shells. In addition, the programmer should know conventional programming languages like C, Pascal, FORTRAN and Basic.
- The ***project manager*** is the leader of the expert system development team, responsible for keeping the project on track. He or she makes sure that all deliverables and milestones are met, interacts with the expert, knowledge engineer, programmer and end-user.
- The ***end-user***, often called just the *user*, is a person who uses the expert system when it is developed. The user must not only be confident in the expert system performance but also feel comfortable using it. Therefore, the design of the user interface of the expert system is also vital for the project's success; the end-user's contribution here can be crucial.

Structure of a rule-based expert system

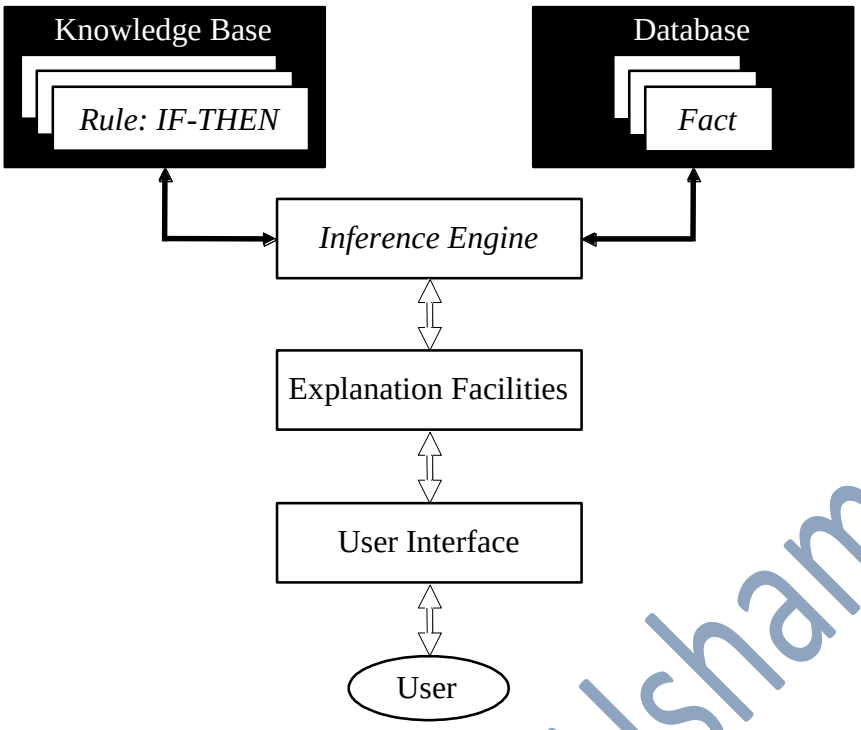
- The production model is based on the idea that humans solve problems by applying their knowledge (expressed as production rules) to a given problem represented by problem-specific information.

- The production rules are stored in the long-term memory and the problem-specific information or facts in the short-term memory.

Production system model



Basic structure of a rule-based expert system



- The **knowledge base** contains the domain knowledge useful for problem solving. In a rule-based expert system, the knowledge is represented as a set of rules. Each rule specifies a relation, recommendation, directive, strategy or heuristic and has the IF (condition) THEN (action) structure. When the condition part of a rule is satisfied, the rule is said to ***fire*** and the action part is executed.
- The **database** includes a set of facts used to match against the IF (condition) parts of rules stored in the knowledge base.

Inference engine

- The **inference engine** carries out the reasoning whereby the expert system reaches a solution. It links the rules given in the knowledge base with the facts provided in the database.

- The inference engine is a generic control mechanism for navigating through and manipulating knowledge and deduce results in an organized manner.

‡ Inference engine the other key component of all expert systems.

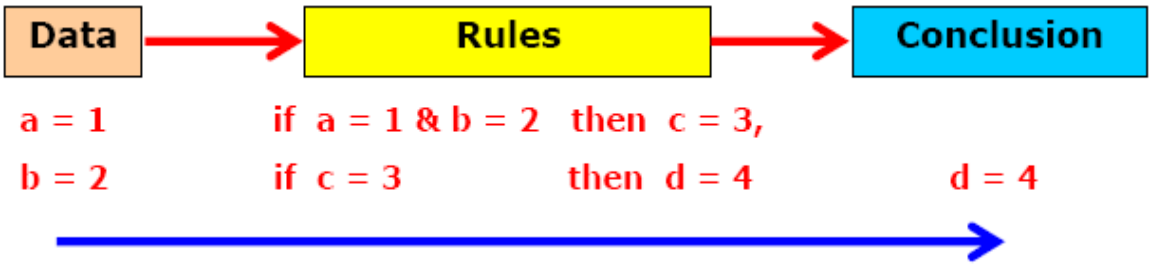
‡ Just a knowledge base alone is not of much use if there are no facilities for navigating through and manipulating the knowledge to deduce something from knowledge base.

‡ 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.

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



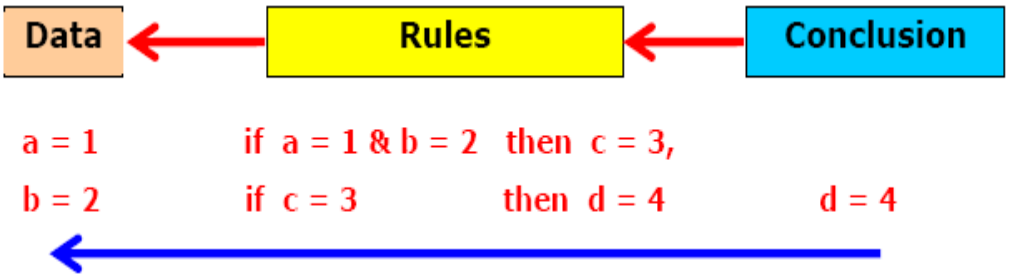
‡ Example : Forward Channing
■ Given : A Rule base contains following Rule set
Rule 1: If A and C Then F
Rule 2: If A and E Then G
Rule 3: If B Then E

Rule 4: If G Then D
■ Problem : Prove
If A and B true Then D is true

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

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

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

Searches
So a knowledge base may be represent as a branching network or tree. Many tree searching algorithms exists but two basic approaches are depth-first search and breadth-first search.

The **explanation**

- A unique feature of an expert system is its **explanation capability**. It enables the expert system to review its own reasoning and explain its decisions.
- The **explanation facilities** enable the user to ask the expert system ***how*** a particular conclusion is reached and ***why*** a specific fact is needed. An expert system must be able to explain its reasoning and justify its advice, analysis or conclusion.
- **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;

User Interface
The **user interface** is the means of communication between a user seeking a solution to the problem and an expert system.
■ **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.

Can expert systems make mistakes?

- Even a brilliant expert is only a human and thus can make mistakes. This suggests that an expert system built to perform

at a human expert level also should be allowed to make mistakes. But we still trust experts, even we recognise that their judgements are sometimes wrong. Likewise, at least in most cases, we can rely on solutions provided by expert systems, but mistakes are possible and we should be aware of this.

How do we choose between forward and backward chaining?

- If an expert first needs to gather some information and then tries to infer from it whatever can be inferred, choose the forward chaining inference engine.
- However, if your expert begins with a hypothetical solution and then attempts to find facts to prove it, choose the backward chaining inference engine.

Advantages of rule-based expert systems

- **Natural knowledge representation**. An expert usually explains the problem-solving procedure with such expressions as this: "In such-and-such situation, I do so-and-so". These expressions can be represented quite naturally as IF-THEN production rules.
- **Uniform structure**. Production rules have the uniform IF-THEN structure. Each rule is an independent piece of knowledge. The very syntax of production rules enables them to be self-documented.
- **Separation of knowledge from its processing**. The structure of a rule-based expert system provides an effective separation of the knowledge base from the inference engine. This makes it possible to develop different applications using the same expert system shell.
- ***Dealing with incomplete and uncertain knowledge***. Most rule-based expert systems are capable of representing and reasoning with incomplete and uncertain knowledge.