Introduction to Expert Systems

What is an expert system?

"An expert system is a computer system that emulates, or acts in all respects, with the decision-making capabilities of a human expert."

Professor Edward Feigenbaum

Stanford University

What is an expert system?

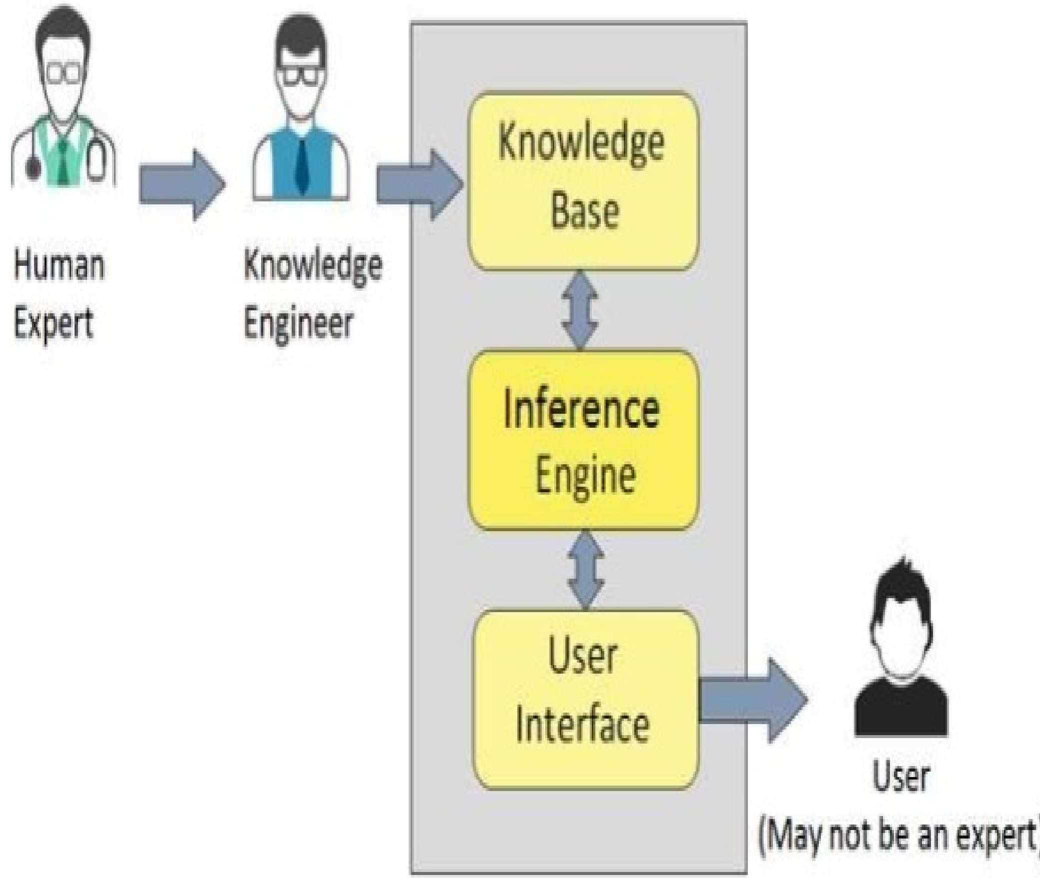
- The expert systems are the computer applications developed to solve complex problems in a particular domain, at the level of extra-ordinary human intelligence and expertise.
- Characteristics of Expert Systems
- High performance
- Understandable
- Reliable
- Highly responsive

Capabilities of Expert Systems

- Advising
- Instructing and assisting human in decision making
- Demonstrating
- Deriving a solution
- Diagnosing
- Explaining
- Interpreting input
- Predicting results
- Justifying the conclusion
- Suggesting alternative options to a problem

Components of Expert Systems

- 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…

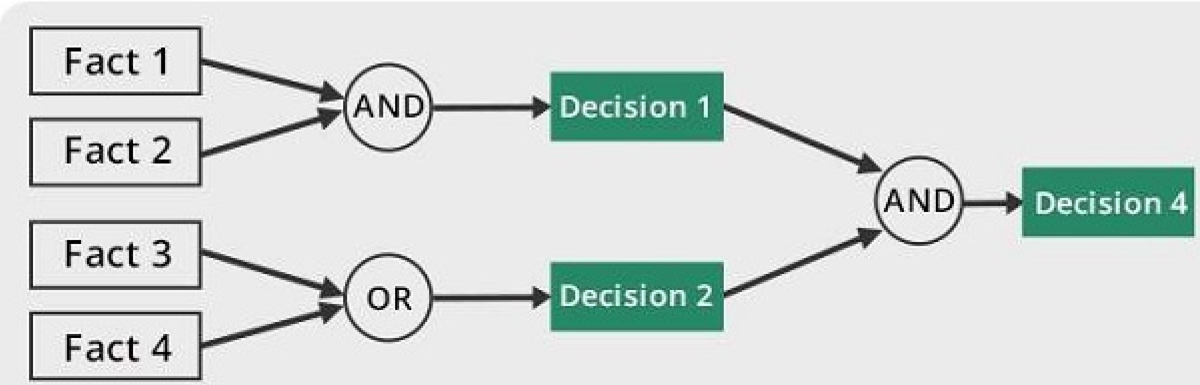
- **Knowledge representation**
- It is the method used to organize and formalize the knowledge in the knowledge base. It is in the form of IF-THEN-ELSE rules.
- **Knowledge Acquisition**
- The success of any expert system majorly depends on the quality, completeness, and accuracy of the information stored 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, flawless 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

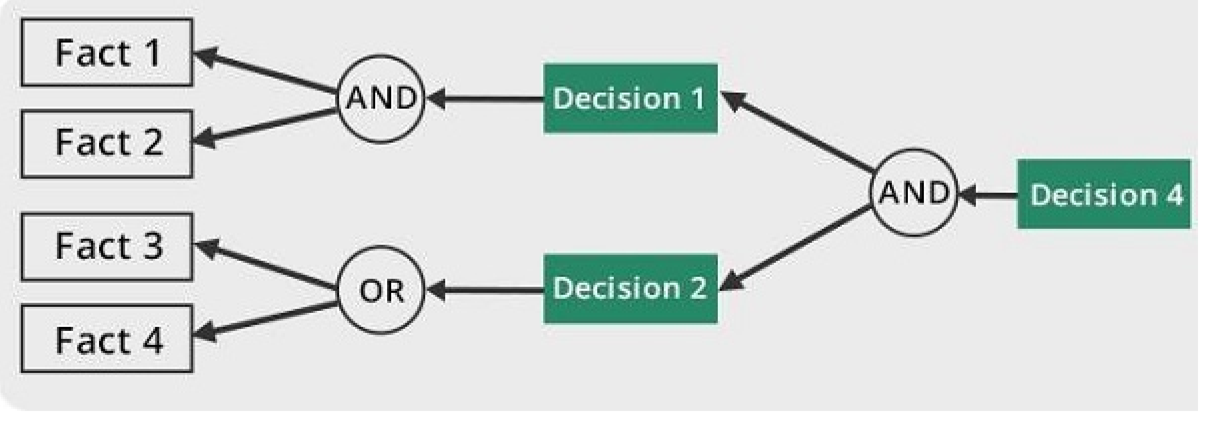
- It is a strategy of an expert system to answer the question, “What can happen next?”
- Here, the Inference Engine follows the chain of conditions and derivations and finally deduces the outcome. It considers all the facts and rules, and sorts them before concluding to a solution.
- This strategy is 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?”

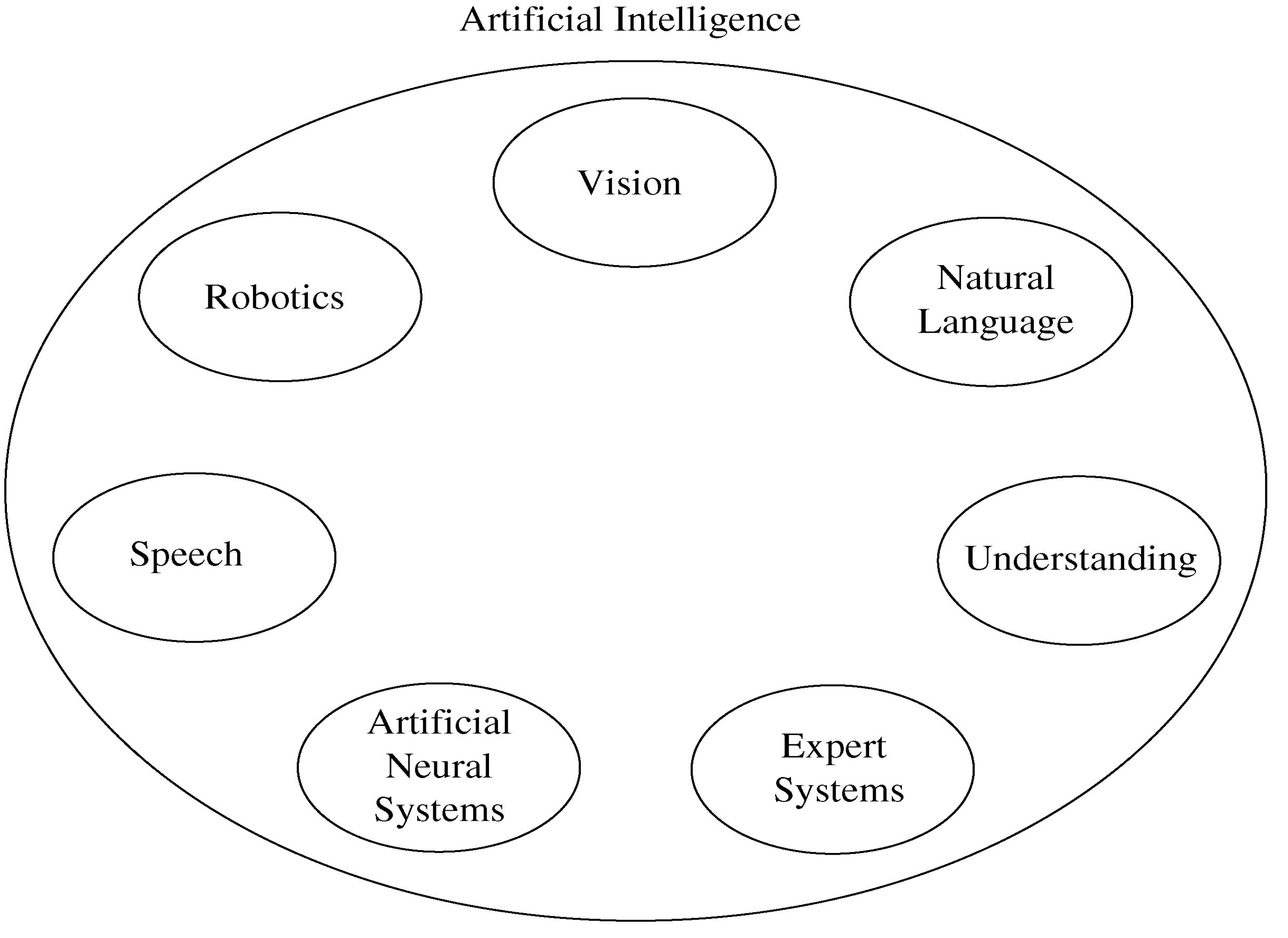
- On the basis of what has already happened, the Inference Engine tries to find out which conditions could have happened in the past for this result. This strategy is followed for finding out cause or reason. For example, diagnosis of blood cancer in humans.



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.

Areas of Artificial Intelligence



Limitations of Expert Systems

- Typical expert systems cannot generalize through analogy to reason about new situations in the way people can.

- A knowledge acquisition bottleneck results from the time-consuming and labor intensive task of building an expert system.

Early Expert Systems

- DENDRAL – used in chemical mass spectroscopy to identify chemical constituents
- MYCIN – medical diagnosis of illness
- DIPMETER – geological data analysis for oil
- PROSPECTOR – geological data analysis for minerals
- XCON/R1 – configuring computer systems

Problems with Algorithmic Solutions

- Conventional computer programs generally solve problems having algorithmic solutions.
- Algorithmic languages include C, Java, and C#.
- Classical AI languages include LISP and PROLOG.

Considerations for Building Expert Systems

- Can the problem be solved effectively by conventional programming?
- Is there a need and a desire for an expert system?
- Is there at least one human expert who is willing to cooperate?
- Can the expert explain the knowledge to the knowledge engineer can understand it.
- Is the problem-solving knowledge mainly heuristic and uncertain?

Languages, Shells, and Tools

- Expert system languages are post-third generation.
- Procedural languages (e.g., C) focus on techniques to represent data.
- More modern languages (e.g., Java) focus on data abstraction.
- Expert system languages (e.g. CLIPS) focus on ways to represent knowledge.

Expert systems Vs conventional programs I

Characteristic	Conventional Program	Expert System
Control by …	Statement order	Inference engine
Control & Data	Implicit integration	Explicit separation
Control Strength	Strong	Weak
Solution by …	Algorithm	Rules & Inference
Solution search	Small or none	Large
Problem solving	Algorithm	Rules

Expert systems Vs conventional programs II

Characteristic	Conventional Program	Expert system
Input	Assumed correct	Incomplete, incorrect
Unexpected input	Difficult to deal with	Very responsive
Output	Always correct	Varies with the problem
Explanation	None	Usually
Applications	Numeric, file & text	Symbolic reasoning
Execution	Generally sequential	Opportunistic rules

Expert systems Vs conventional programs III

Characteristic	Conventional Program	Expert System
Program Design	Structured design	Little or no structure
Modifiability	Difficult	Reasonable
Expansion	Done in major lumps	Incremental

Elements of an Expert System

- User interface – mechanism by which user and system communicate.
- Exploration facility – explains reasoning of expert system to user.
- Working memory – global database of facts used by rules.
- Inference engine – makes inferences deciding which rules are satisfied and prioritizing.

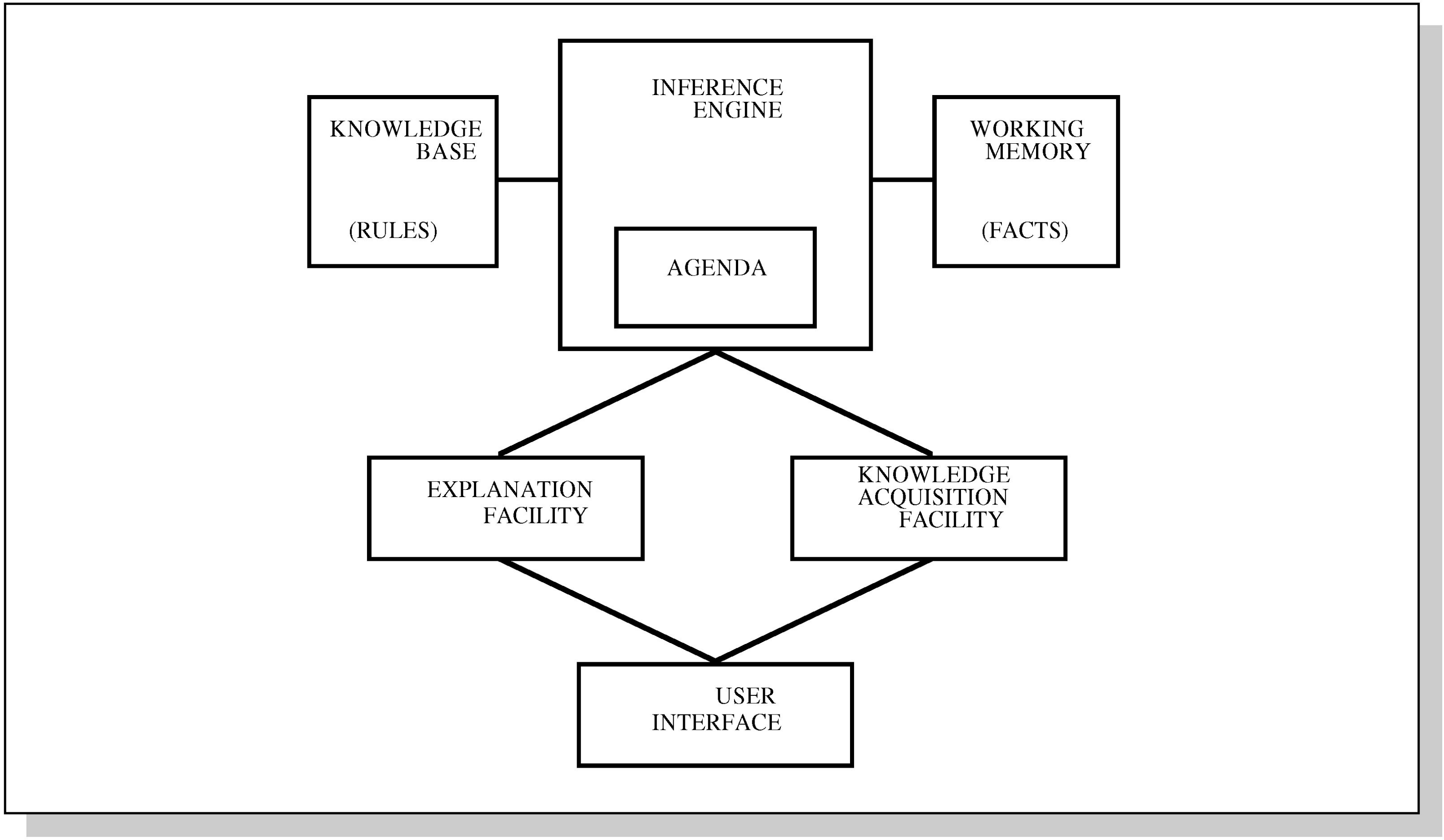
Elements Continued

- Agenda – a prioritized list of rules created by the inference engine, whose patterns are satisfied by facts or objects in working memory.
- Knowledge acquisition facility – automatic way for the user to enter knowledge in the system bypassing the explicit coding by knowledge engineer.
- Knowledge Base – includes the rules of the expert system

Production Rules

- Knowledge base is also called production memory.
- Production rules can be expressed in IF-THEN pseudocode format.
- In rule-based systems, the inference engine determines which rule antecedents are satisfied by the facts.

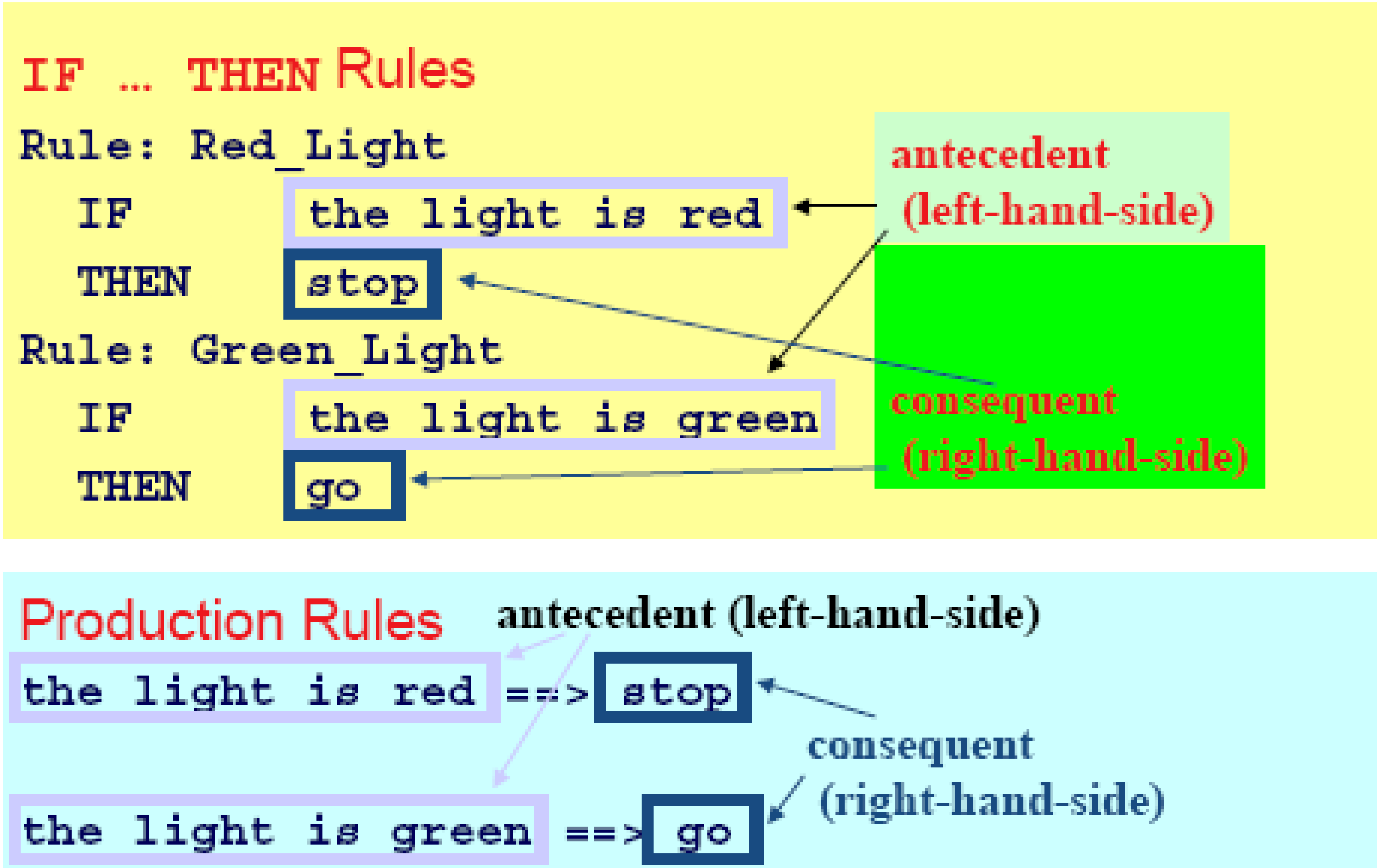
Structure of a Rule-Based Expert System



Rule-Based ES

- knowledge is encoded as `IF … THEN` rules
  – these rules can also be written as *production rules*
- the inference engine determines which rule antecedents are satisfied
  – the left-hand side must "match" a fact in the working memory
- satisfied rules are placed on the agenda
- rules on the agenda can be activated ("fired")
  – an activated rule may generate new facts through its right-hand side
  – the activation of one rule may subsequently cause the activation of other rules

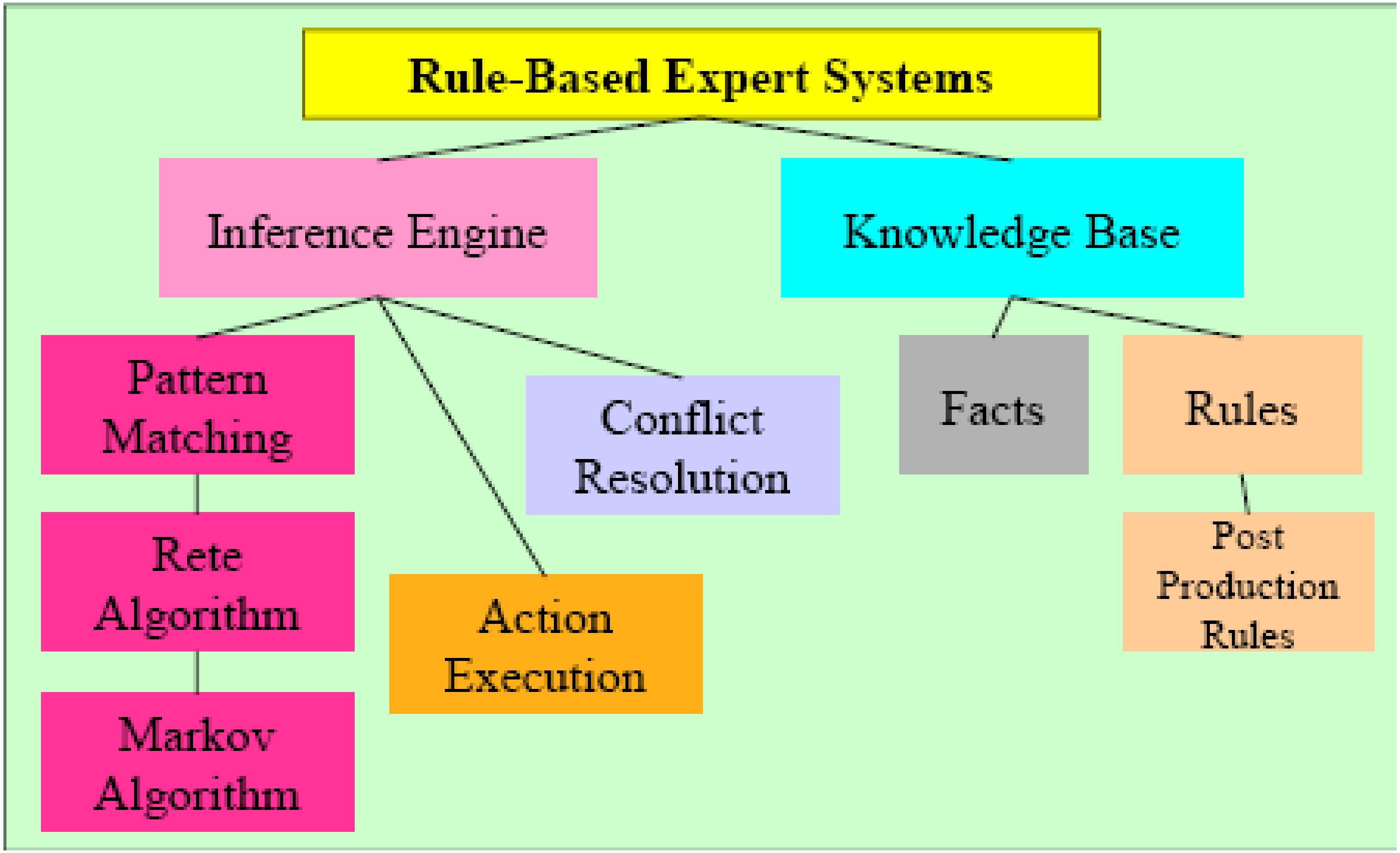
Example Rules



Inference Engine Cycle

- The inference engine determines the execution of the rules by the following cycle:
  - conflict resolution
    - select the rule with the highest priority from the agenda
  - execution (Act)
    - perform the actions on the consequent of the selected rule
    - remove the rule from the agenda
  - match
    - update the agenda
      - add rules whose antecedents are satisfied to the agenda
      - remove rules with non-satisfied agendas
- the cycle ends when no more rules are on the agenda, or when an explicit stop command is encountered

Foundation of Expert Systems



General Methods of Inferencing

- Forward chaining (data-driven)– reasoning from facts to the conclusions resulting from those facts – best for prognosis, monitoring, and control.
  - Examples: CLIPS, OPS5
- Backward chaining (query/Goal driven)– reasoning in reverse from a hypothesis, a potential conclusion to be proved to the facts that support the hypothesis – best for diagnosis problems.
  - Examples: MYCIN