2018-2019 ACADEMIC YEAR, FIRST SEMESTER
LECTURE NOTE FOR 5TH YEAR COE STUDENT

RULE-BASED EXPERT SYSTEMS

IT-51033 Artificial Intelligence

Content

- ✓ Introduction, or What Is Knowledge?
- ✓ Rules As A Knowledge Representation Technique
- ✓ The Main Players In The Development Team
- ✓ Structure of A Rule-based Expert System
- ✓ Characteristics of An Expert System
- ✓ Forward Changing And Backward Changing
- ✓ Conflict Resolution

Introduction

- **What is Knowledge?**
  - is a theoretical or practical understanding of a subject or a domain.
  - Those who possess knowledge are called experts.

- **Who is Generally Acknowledged as An Expert?**
  - Anyone can be considered a domain expert if he or she has deep knowledge and strong practical experience in a particular domain **(E.g., experts in Electrical machine and life insurance marketing)**
  - An expert is skillful person who can do things other people cannot.

Introduction

- **How do Experts Think?**
  - 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

  - These statements represented in the IF-THEN form are called production rules or just rules.

Introduction

- **Expert System**

A computer program capable of performing at the level of human expert in a narrow problem area.

- **Expert System Shell**

An expert system with the knowledge removed; all users have to do is to add the knowledge in the form of rule and provide relevant data to solve problem

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.
- Rule are relatively easy to create and understand.

Rules as a Knowledge Representation Technique

- Any rules 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>*

Rules as a Knowledge Representation Technique

- The antecedent of a rule incorporates two parts: an object (linguistic object) and its value. The object and its value are linked by an operator.
- The operator identifies the object and assigns the value. Operators such as is, are, is not, are not are used to assign a symbolic value to a linguistic object.
- Expert systems can also used mathematical operators to define an object as numerical and assign it to the numerical value.

IF ‘age of the customer’ < 18

AND ‘cash withdrawal’ > 1000

THEN ‘signature of the parent’ is required

Rules as a Knowledge Representation Technique

- Rules can represent:
  - **Relation:**

    IF the 'fuel tank' is empty
    THEN the car is dead
  - **Recommendation:**

    IF the season is autumn
    AND the sky is cloudy
    AND the forecast is drizzle
    THEN the advice is 'take an umbrella'
  - **Directive:**

    IF the car is dead
    AND the 'fuel tank' is empty
    THEN the action is 'refuel the car'

Rules as a Knowledge Representation Technique

- Rules can represent:
  - **Strategy:**

    IF the car is dead
    THEN the action is 'check the fuel tank' ;
    step 1 is complete
    IF step 1 is complete
    AND the 'fuel tank' is full;
    THEN the action is 'check the battery' ;
    step 2 is complete

  - **Heuristics:**

    IF the spill is liquid
    AND the 'spill pH' < 6
    AND the 'spill smell' is vinegar
    THEN the 'spill material' is 'acetic acid'

The Main Players In The Development Team

- There are five members of the expert system development team:

  1. Domain expert

  2. Knowledge engineer

  3. Programmer

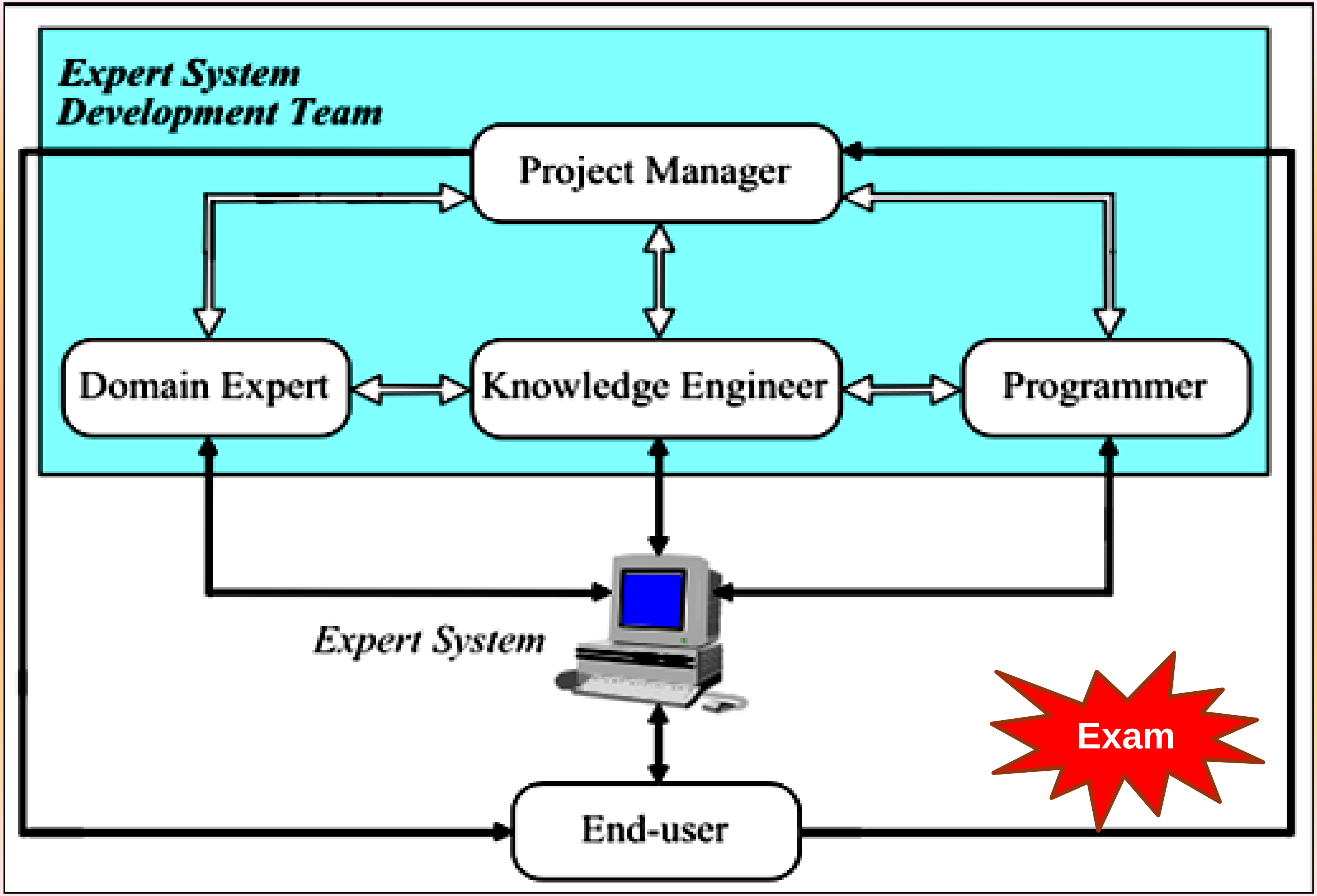
  4. Project manager

  5. End-user

- The success of their expert systems entirely depend on how well the members work together.

Exam

The Main Players In The Development Team



The Main Players In The Development Team

- **Domain Expert:**
  - is a knowledgeable and skilled person capable of solving problems in a specific area or domain and the person's expertise is to be captured in the expert system
  - is the most important person in the expert system development team

Exam

- **Knowledge Engineer:**
  - is someone who is capable of designing, building and testing an expert system into workplace
  - interviews the domain expert to find out how a particular problem is solved

The Main Players In The Development Team

- **Programmer:**
  - is the person responsible for the actual programming, describing the domain knowledge in terms that a computer can understand.

- **Project Manager:**
  - is the leader of the expert system development team, responsible for keeping the project on track
  - makes sure that all deliverables and milestones are met

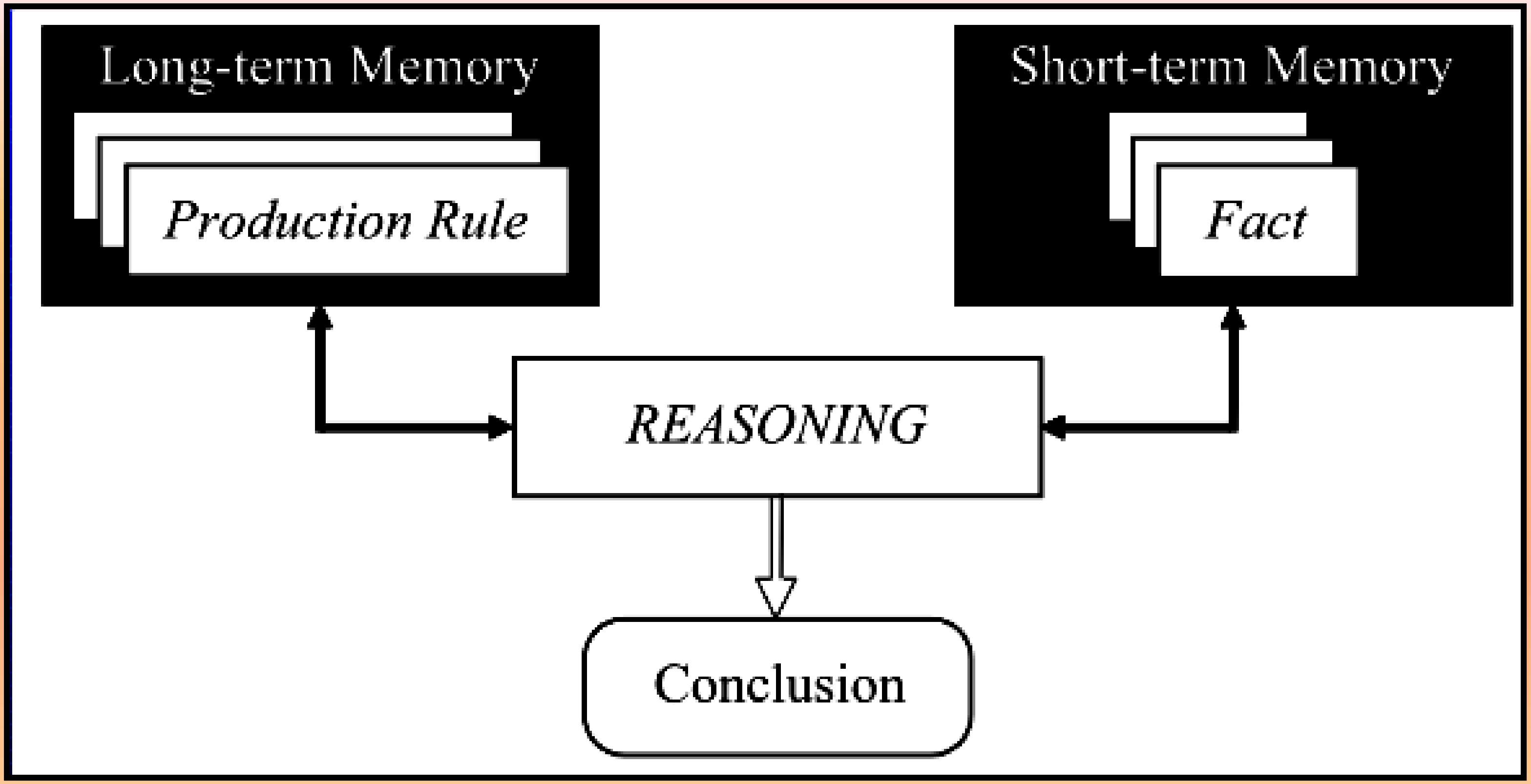
Exam

- **End-user:**
  - often called the user
  - is a person who uses the expert system when it is developed

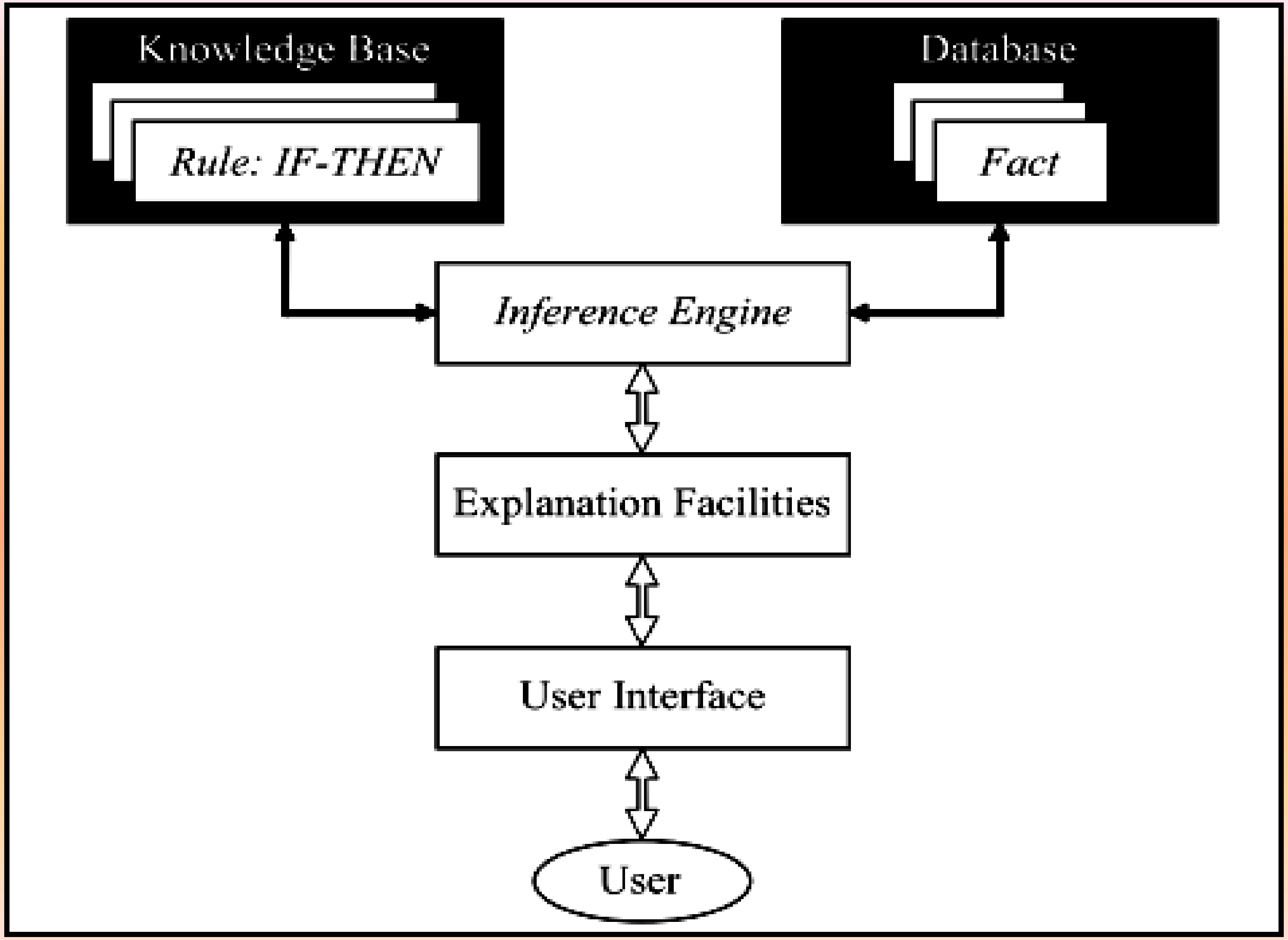
Structure of A Rule-based Expert System

- In early 1970s, Newell and Simon from Carnegie-Mellon University proposed a production system model, the foundation of the modern rule-based expert systems.
- 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



Structure of A Rule-based Expert System

- **Knowledge base**
  - contains the **domain knowledge** useful for problem solving
  - In 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.

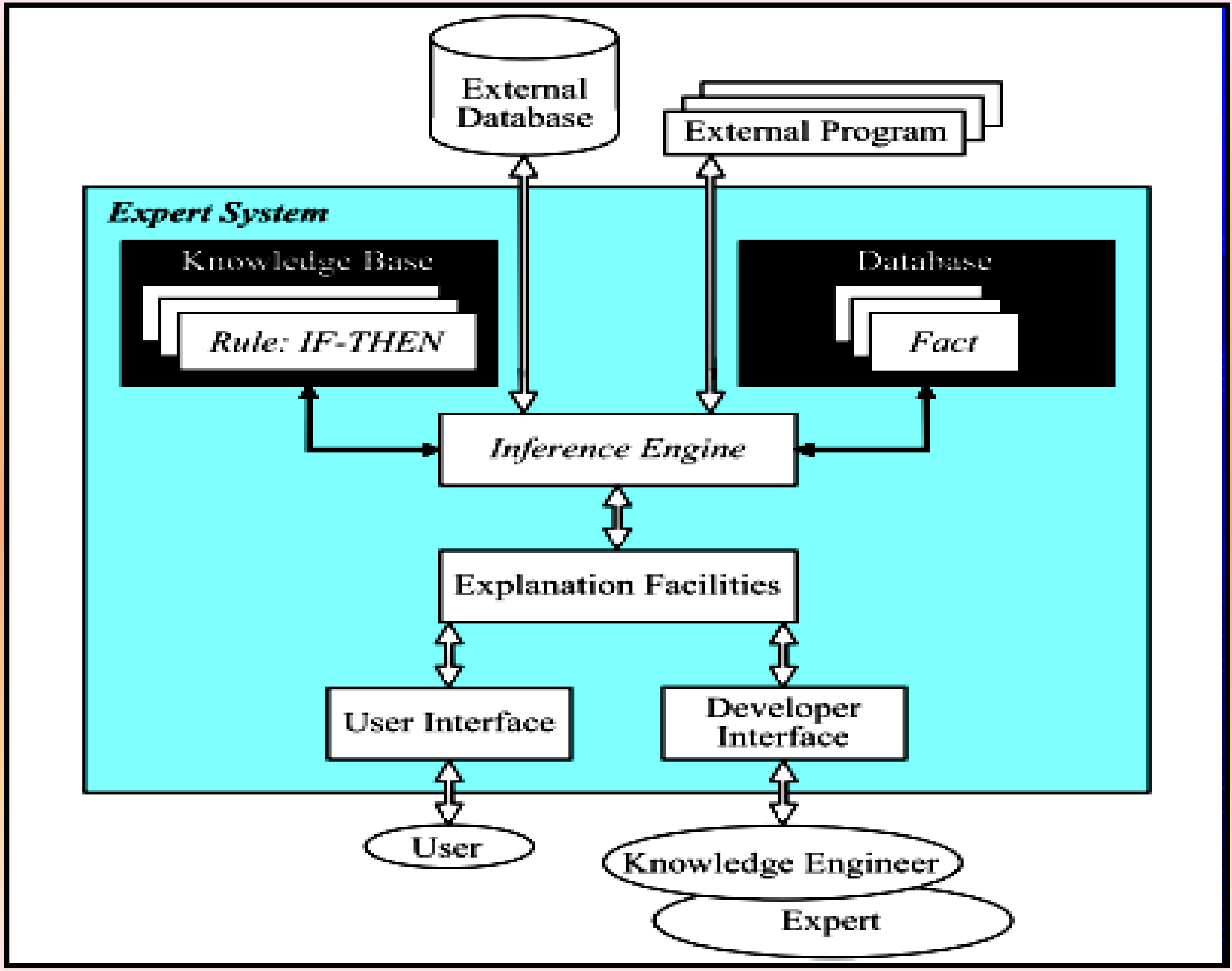
Structure of A Rule-based Expert System

- **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 explanation facilities** enable the user to ask the expert system how a particular conclusion is reached and why a specific fact is needed.

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

Complete Structure of A Rule-based Expert System



Complete Structure of A Rule-based Expert System

- **The external interface** allows an expert system to work with external files and programs written in conventional programming language such as C, Pascal etc.
- **The developer interface** usually includes knowledge base editors, debugging aids and input/output facilities.
  - **knowledge base editors:**

    - text editor: to input and modify rules, to check correct format and spelling.

    - book-keeping facilities: to monitor the changes made by the knowledge engineer or expert. If a rule is changed, the editor will automatically store the change (date and the name of the person) for later reference.

Complete Structure of A Rule-based Expert System

- **Debugging aids** usually consist of tracing facilities and break packages.
- **Tracing facilities** provides a list of all rules fired during the program's execution, and **a break package** makes it possible to tell the expert system in advance where to stop so that the knowledge engineer or the expert can examine the current values in the database.
- Most expert systems also accommodate **input/output facilities** such as run-time knowledge acquisition. This enables the running expert system to ask for needed information whenever this information is not available in the database. When the requested information is input by the knowledge engineer or the expert, the program resumes.

Fundamental Characteristics of An Expert System

- The most important characteristic of an expert system is its high-quality performance.
- No matter how fast the system can solve a problem, the user will not be satisfied if the result is wrong.
- On the other hand, the speed of reaching a solution is very important.
- Even the most accurate decision or diagnosis may not be useful if it is too late to apply, for instance, in an emergency, when a patient dies or a nuclear power plant explodes.
- Like human experts, expert systems should apply heuristics to guide the reasoning and thus reduce the search area for a solution.

Fundamental Characteristics of An Expert System

- The explanation capability enables the expert system to review its own reasoning and explain its decisions.
- Expert systems employ symbolic reasoning when solving a problem.
- Symbols are used to represent different types of knowledge such as facts, concepts and rules.

Can Expert Systems Make Mistakes?

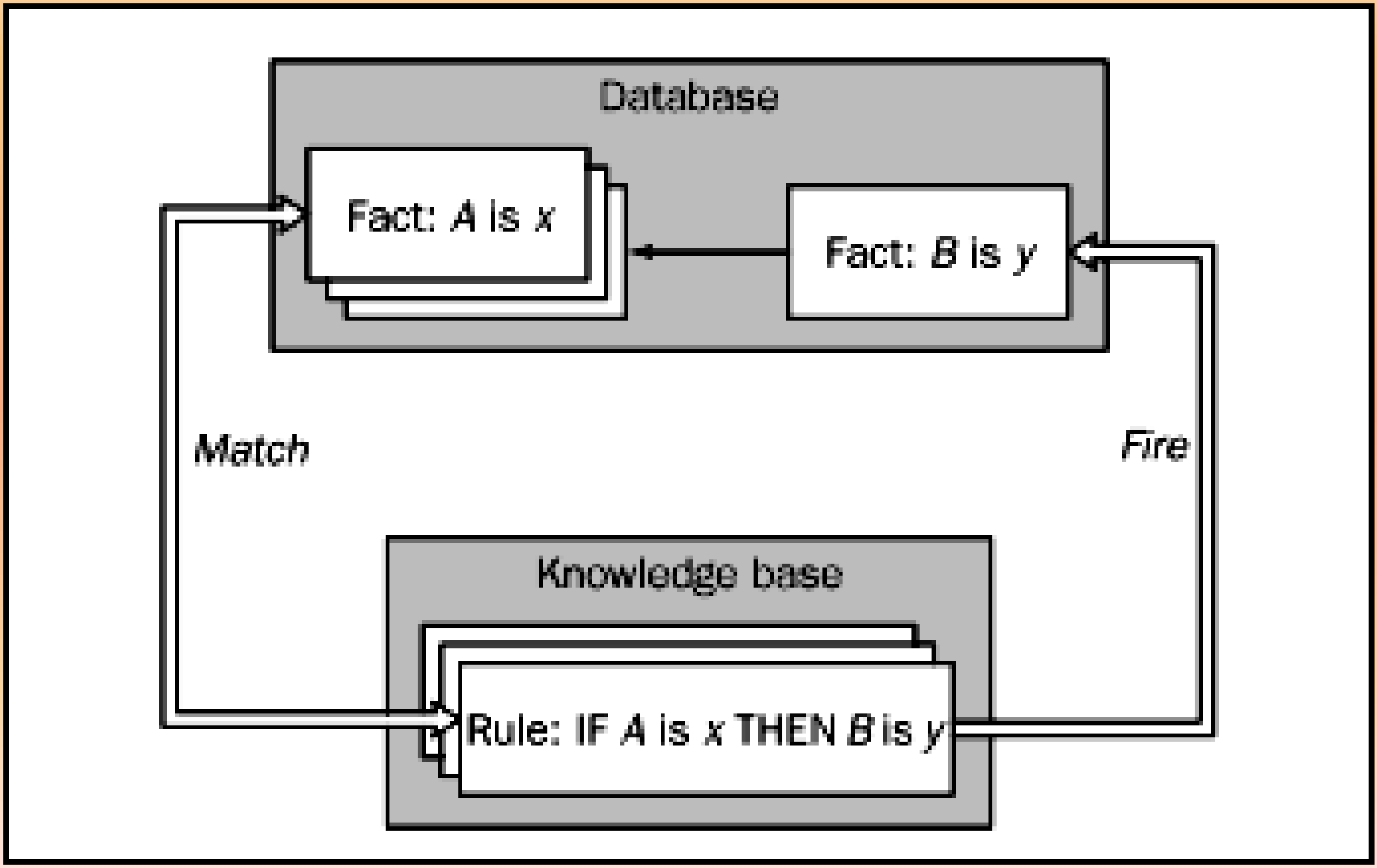
- Even a brilliant expert is only a human and thus can make mistakes.
- This suggests that an expert system built to perform at a human expert level also should be allowed to make mistakes.

Forward Chaining And Backward Chaining Inference Techniques

- In a rule-based expert system, the domain knowledge is represented by a set of IF-THEN production rules and 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. The fired rule may change the set of facts by adding a new fact.
- Letters in the database and the knowledge base are used to represent situations or concepts.
- The matching of the rule IF parts to the facts produces inference chains.

Inference Engine Cycles Via A Match-Fire Procedure

- The inference chain indicates how an expert system applies the rules to reach a conclusion.

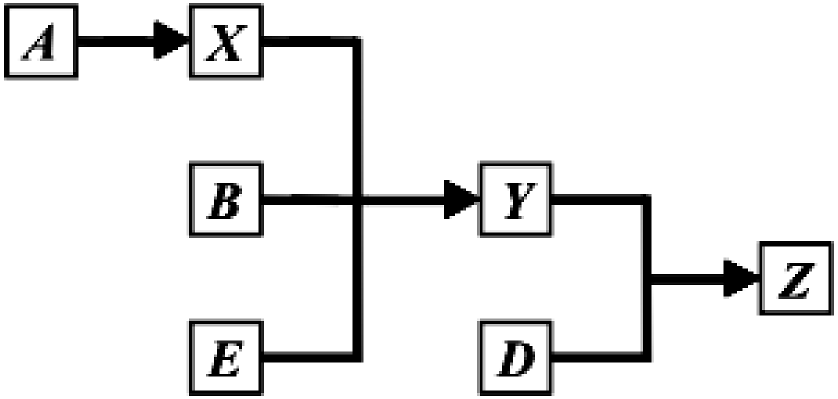


An Example of An Inference Chain

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

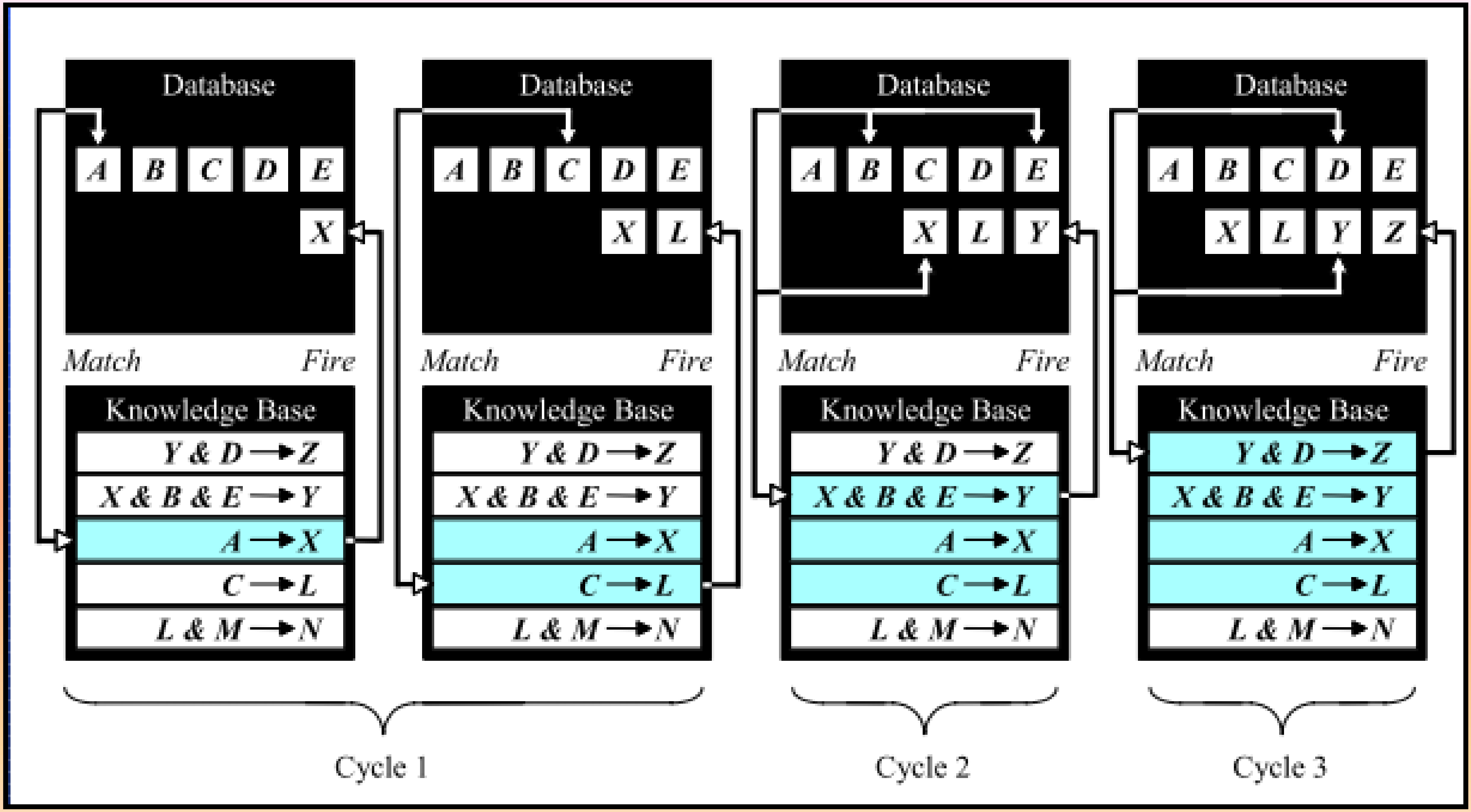
Rule 3: IF *A* is true
THEN *X* is true



Forward Chaining

- Forward chaining is the data-driven reasoning.
- The reasoning starts from the known data and proceeds forward with that data until it reaches the goal.
- Each time only the topmost rule is executed.
- When fired, the rule adds a new fact in the database.
- Any rule can be executed only once.
- The match-fire cycle stops when no further rules can be fired.

Forward Chaining



Forward Chaining

- Forward chaining is a technique for gathering information and then inferring from it whatever can be inferred.
- However, in forward chaining, many rules may be executed that have nothing to do with the established goal.
- Therefore, if our goal is to infer only one particular fact, the forward chaining inference technique would not be efficient.

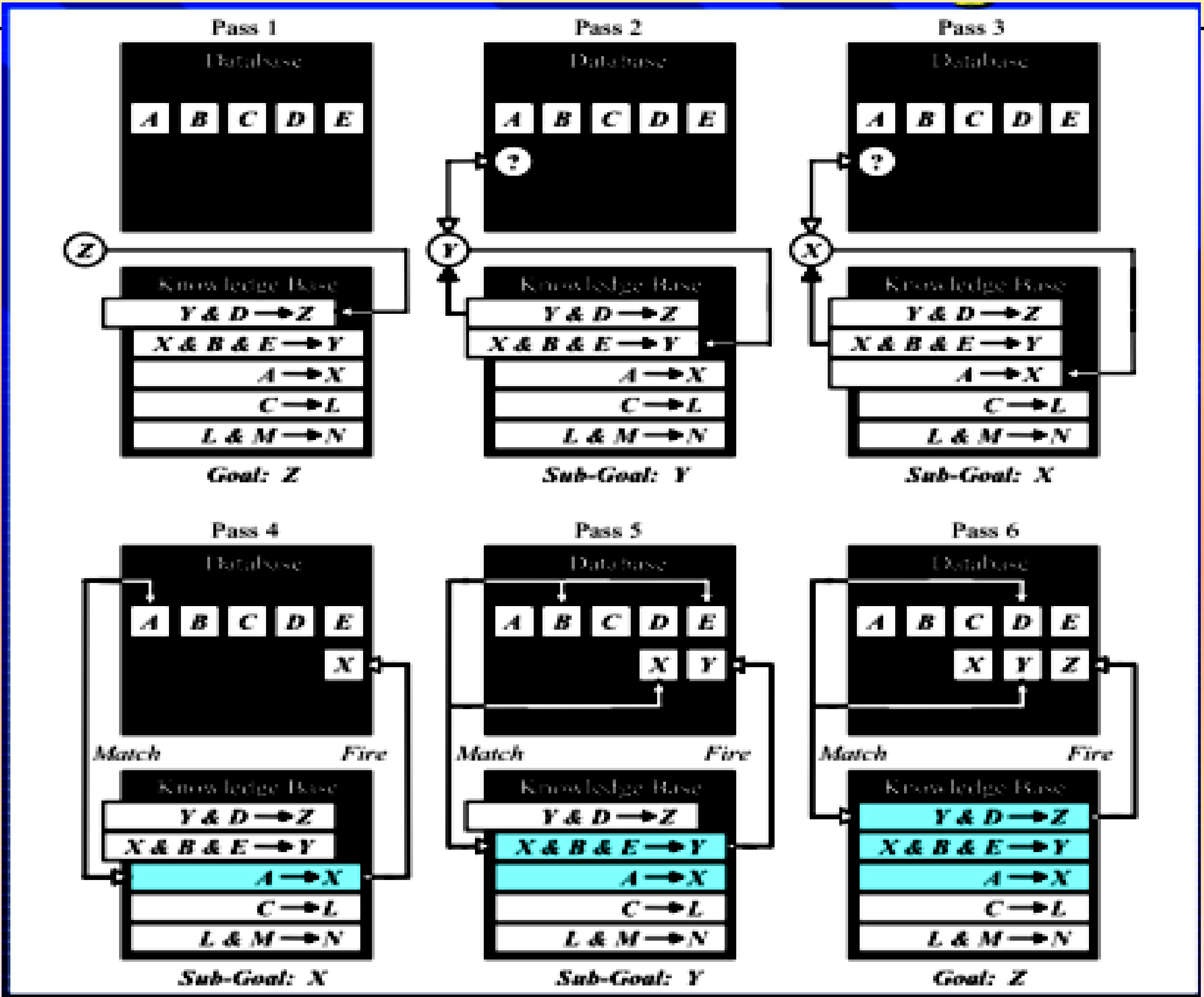
Backward Chaining

- Backward chaining is the goal-driven reasoning.
- In backward chaining, an expert system has the goal (a hypothetical solution) and the inference engine attempts to find the evidence to prove it.
- First, the knowledge base is searched to find rules that might have the desired solution.
- Such rules must have the goal in their THEN (action) parts.
- If such a rule is found and its IF (condition) part matches data in the database, then the rule is fired and the goal is proved.
- However, this is rarely the case.

Backward Chaining

- Thus the inference engine puts aside the rule it is working with (the rule is said to stack) and sets up a new goal, a sub-goal, to prove the IF part of this rule.
- Then the knowledge base is searched again for rules that can prove the sub-goal.
- The inference engine repeats the process of stacking the rules until no rules are found in the knowledge base to prove the current sub-goal.

Backward Chaining



How Do We Choose Between Forward And Backward Chaining?

- Forward chaining example, DENDRAL, an expert system for determining the molecular structure of unknown soil based on its mass spectral data (Feigenbaumet al., 1971).
- Most backward chaining expert systems are used for diagnostic purposes. For instance, MYCIN, a medical expert system for diagnosing infectious blood diseases (Shortliffe, 1976).

Conflict Resolution

- A method for choosing a rule to fire when more than one rule can be fired in a given cycle is called conflict resolution.

Rule 1:

IF the 'traffic light' is green

THEN the action is go

Rule 2:

IF the 'traffic light' is red

THEN the action is stop

Rule 3:

IF the 'traffic light' is red

THEN the action is go

Conflict Resolution

- In forward chaining, both rules would be fired. Rule 2 is fired first as the top-most one, and as a result, its THEN part is executed and linguistic object action obtains value stop.
- However, Rule 3 is also fired because the condition part of this rule matches the fact 'traffic light' is red, which is still in the database.
- As a consequence, object action takes new value go.
- This simple example shows that the rule order is vital when the forward chaining inference technique is used.

Exam

Methods Used for Conflict Resolution of Rules

- **Fire the rule with the highest priority.** In simple applications, the priority can be established by placing the rules in an appropriate order in the knowledge base. Usually this strategy works well for expert systems with around 100 rules.
- **Fire the most specific rule.** This method is also known as the longest matching strategy. It is based on the assumption that a specific rule processes more information than a general one.
- **Fire the rule that uses the data most recently entered in the database.** This method relies on time tags attached to each fact in the database. In the conflict set, the expert system first fires the rule whose antecedent uses the data most recently added to the database.

Methods Used for Conflict Resolution

RULE 1 Meningitis Prescription1
(Priority 100)
IF Infection is Meningitis
AND The Patient is a Child
THEN Prescription is Number_1
AND Drug Recommendation is Ampicillin
AND Drug Recommendation is Gentamicin
AND Display Meningitis Prescription1

RULE 2 Meningitis Prescription2
(Priority 90)
IF Infection is Meningitis
AND The Patient is an Adult
THEN Prescription is Number_2
AND Drug Recommendation is Penicillin
AND Display Meningitis Prescription2

Rule 1:
IF the season is autumn
AND the sky is cloudy
AND the forecast is rain
THEN the advice is 'stay home'

Rule 2:
IF the season is autumn
THEN the advice is 'take an umbrella'

Rule 1:
IF the forecast is rain [08:16 PM 11/25/96]
THEN the advice is 'take an umbrella'

Rule 2:
IF the weather is wet [10:18 AM 11/26/96]
THEN the advice is 'stay home'

End of Chapter