Artificial Intelligence

Prof. Dr. Sarjon Defit, S.Kom, MSc

Introduction to Intelligent Systems

- Introduce students to the fundamentals and applications of intelligent systems.
- Includes neural networks, genetic algorithms, fuzzy logic, rough set as well as rule-based and hybrid expert systems
- Methods have been applied successfully to a variety of problems ranging from chess playing and predicting financial markets to detecting cancer cells.

Introduction to Intelligent Systems

- Understand the basic concepts of artificial intelligence
- Understand how the different methodologies can be used to solve complex problems in the real world.
- Ability to evaluate and choose the right intelligent system methodology to solve a particular problem.
- Know when to use intelligent systems for a particular problem.
- Be able to build a simple Expert System and Artificial Neural Network application.

Intelligent Machines

- The *Big Questions* of the Universe:
  - **How does a human mind work**
  - **Can non-humans have minds?**
- ***Intelligence*** is the ability to understand and learn things.
- ***Intelligence*** is the ability to think and understand instead of doing things by instinct or automatically.
- We can define intelligence as ***the ability to learn and understand, to solve problems and to make decisions***.

Rule Based Expert Systems

- What is knowledge?
- Rules as a knowledge representation technique
- Structure of a rule-based expert system
- Characteristics of an expert system
- Forward chaining and backward chaining
- Conflict resolution
- Summary

Knowledge

- **Knowledge** is a theoretical or practical understanding of a subject or a domain.
- Knowledge is also the sum of what is currently known
- 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.

Rules

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

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***).

Rules

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

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

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

Rules

- 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***.
- Operators such as *is*, *are*, *is not*, *are not* are used to assign a **symbolic value** to a linguistic object.
- Mathematical operators define an object as numerical and assign it a **numerical value**.

IF 'age of the customer' $<$ 18
AND 'cash withdrawal' $>$ 1000
THEN 'signature of the parent' is required

Rules: Relations or Recommendations

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’

Directives and Strategies

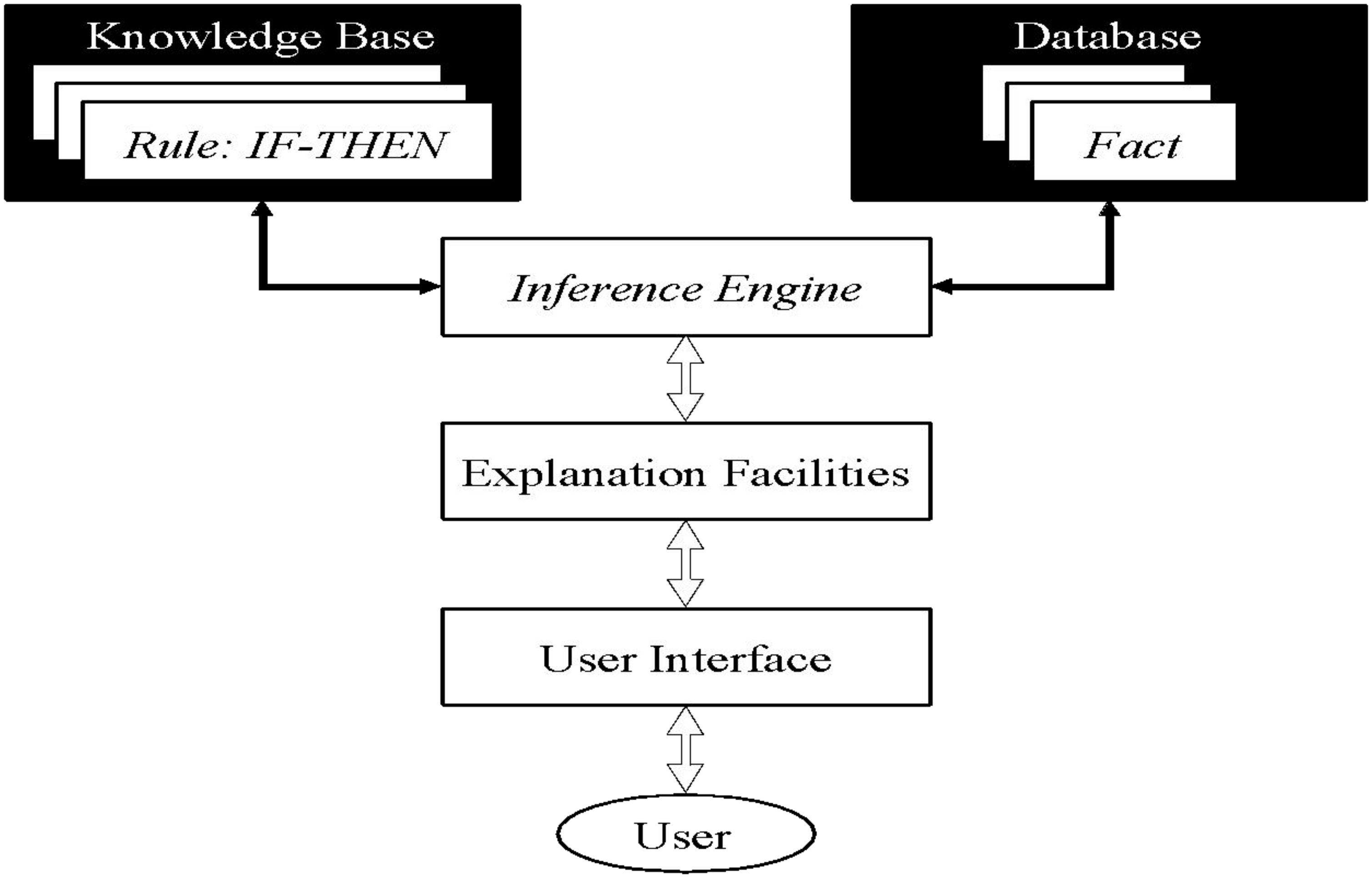
- **Directive**

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

  IF the car is dead
  THEN the action is 'check the fuel tank';
  step1 is complete

  IF step1 is complete
  AND the 'fuel tank' is full
  THEN the action is 'check the battery';
  step2 is complete

Basic structure of a rule-based expert system



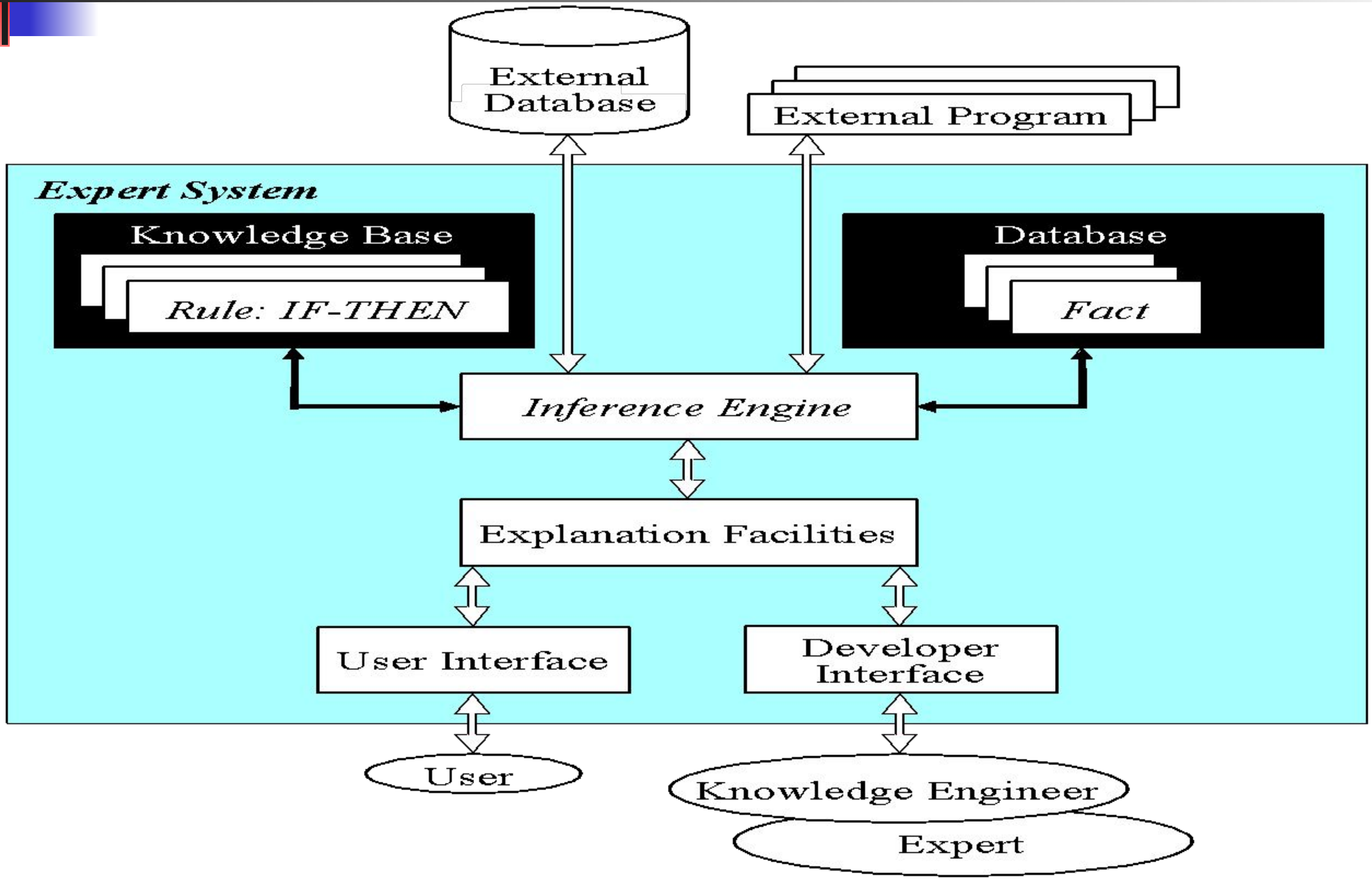
Production Rule Systems

- The **knowledge base** contains the domain knowledge useful for problem solving represented as a set of rules.
- 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.

Production Rule Systems

- The **inference engine** carries out the reasoning. 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 and an expert system.

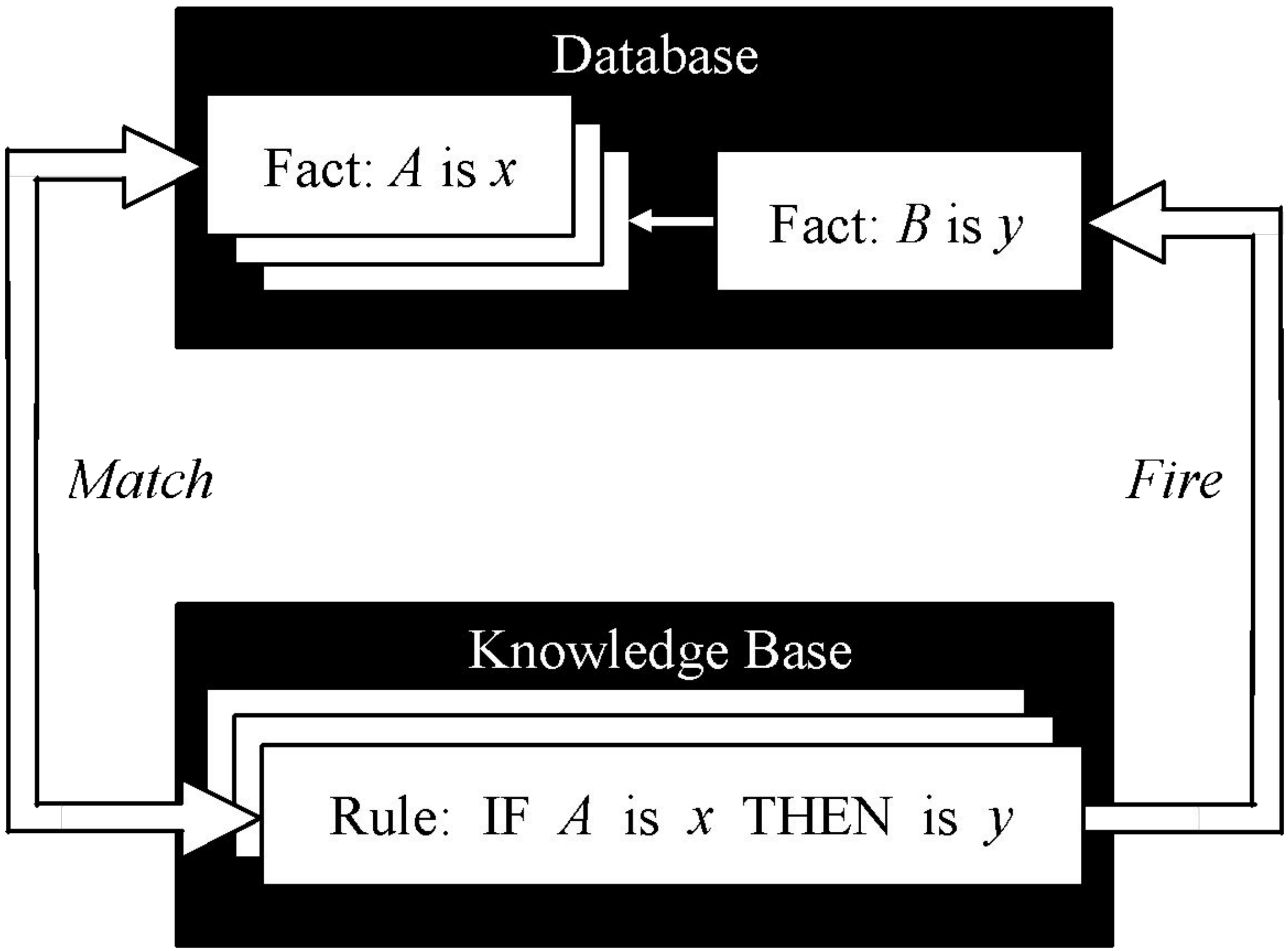
Complete structure of a rule-based expert system



Forward Chaining

- 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.
- The matching of the rule IF parts to the facts produces **inference chains**.
- The inference chain indicates how an expert system applies the rules to reach a conclusion.

Inference engine cycles via a match-fire procedure

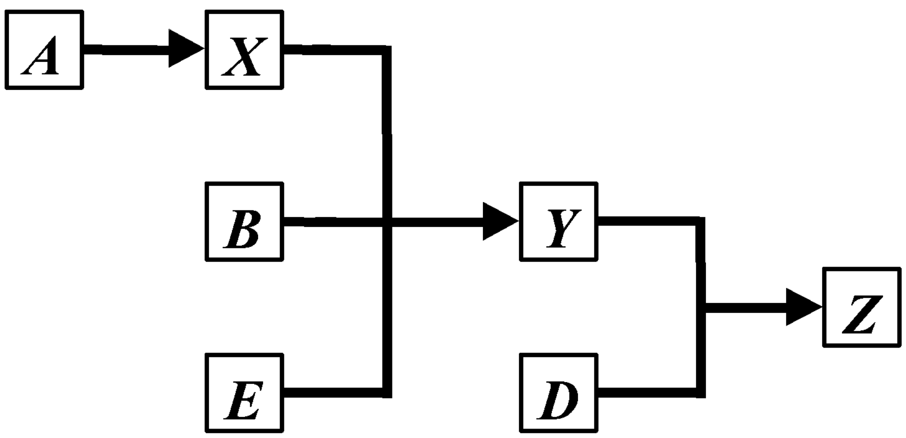


An example of an inference chain

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



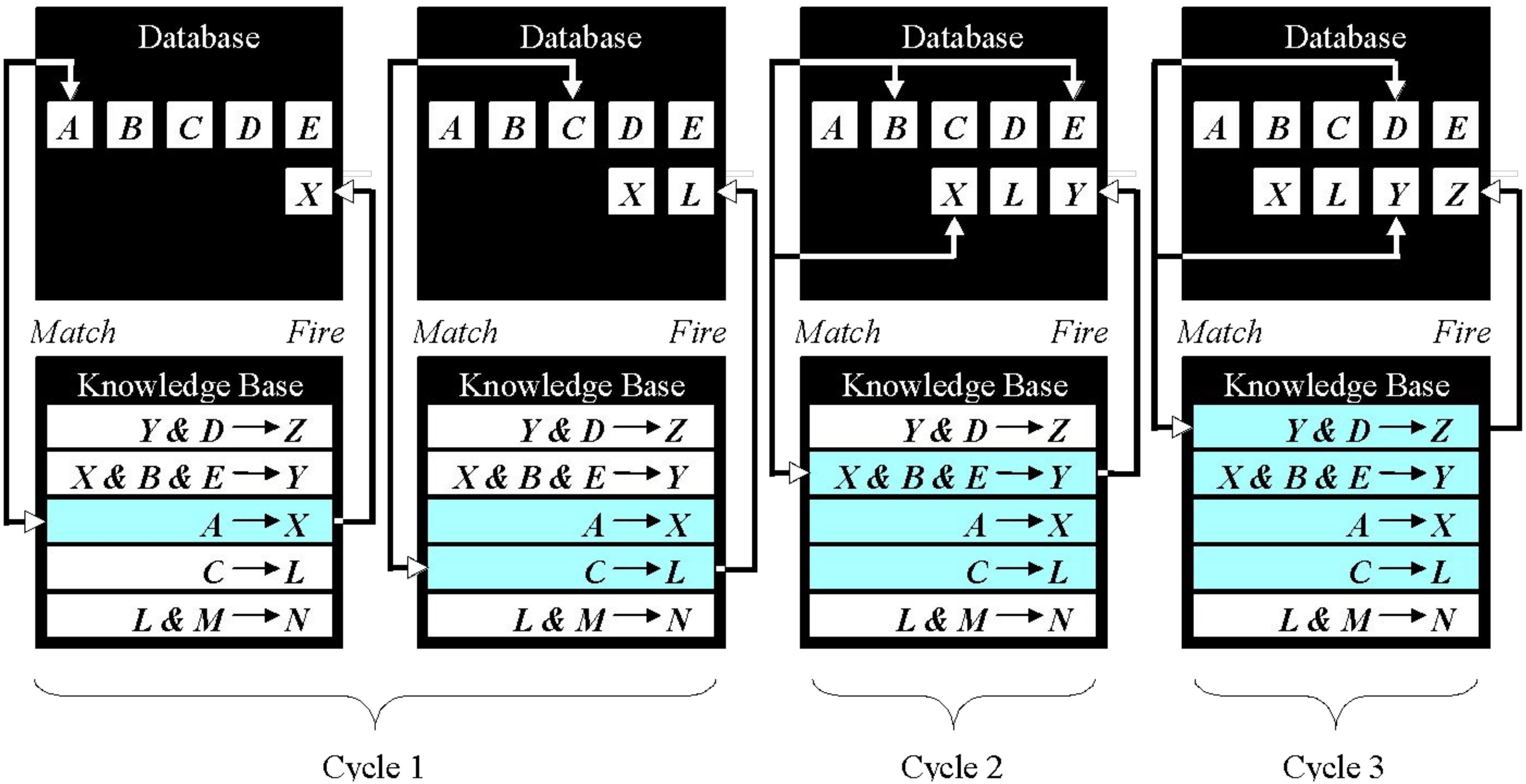
FORWARD CHAINING
VS
BACKWARD CHAINING

Prof. Dr. Sarjon Defit, S.Kom, MSc

Forward Chaining

- Forward chaining is **data-driven reasoning**.
- The reasoning starts from the known data
- The topmost rule is executed and adds a new fact to the database.
- Any rule can be executed only once.
- The match-fire cycle stops when no further rules can be fired

Forward Chaining



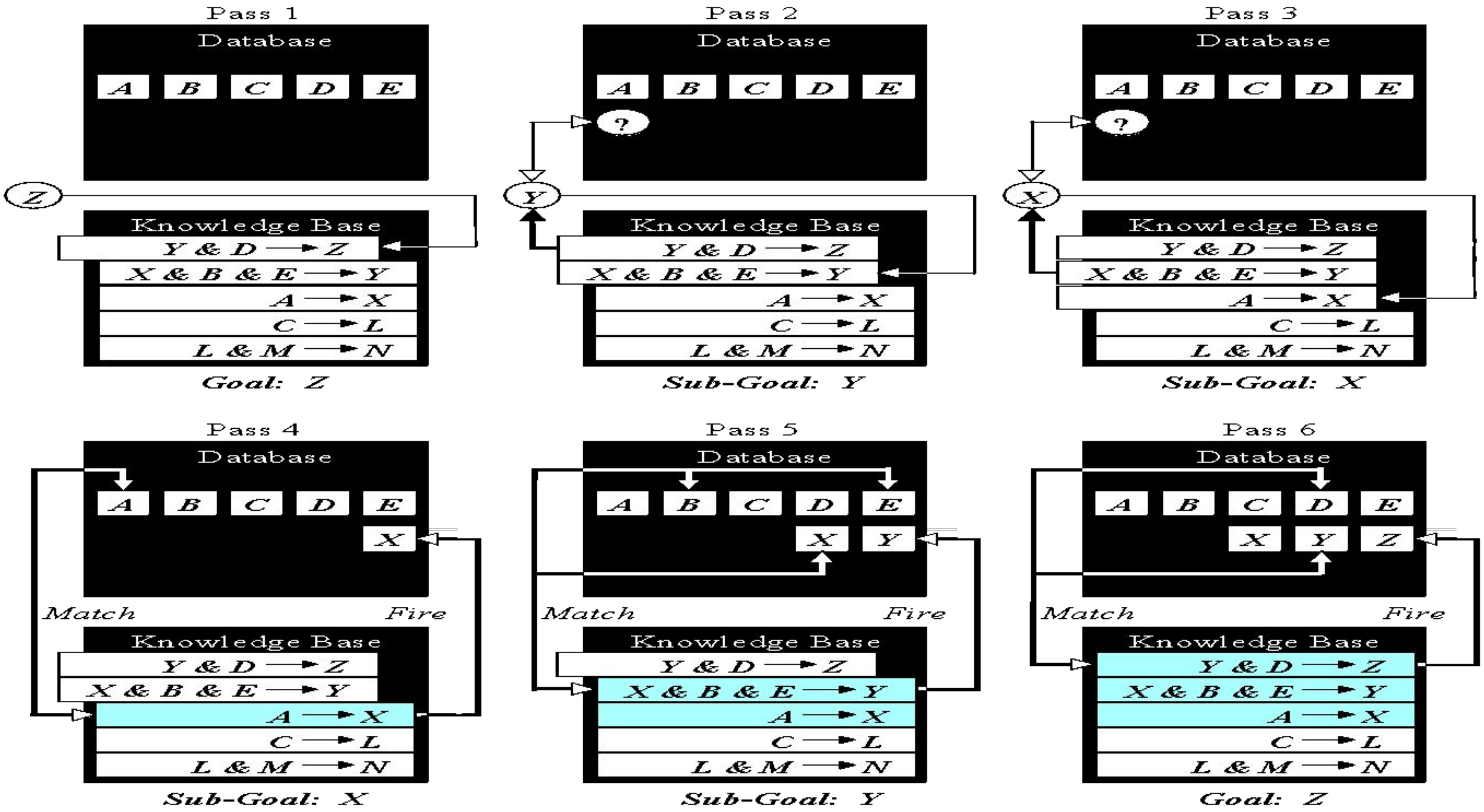
Backward Chaining

- Backward chaining is **goal-driven reasoning**.
- The expert system has the goal (a *hypothetical solution*) and the inference engine attempts to find the evidence to prove it.
- The knowledge base is searched to find rules that might have the desired solution i.e they have the goal in their THEN (action) parts.
- If such a rule is found and its IF (condition) part matches data in the database, the rule is fired and the goal is proved.

Backward Chaining

- If the goal is not proved, 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.
- 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



Forward or 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.

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

Conflict Resolution

- In forward chaining, *BOTH* rules would be fired.
- *Rule* 2 is fired first as the topmost 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*.

Conflict Resolution Methods

- Fire the rule with the ***highest priority***.
- Fire the ***most specific rule***. (***longest matching strategy***.) 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.

Conflict Resolution Methods

Fire the rule with the ***highest priority***.

Goal 1: Prescription is ? Prescription

Rule 1: Meningitis Prescription1 (priority 100)

IF Infection is meningitis and the Patient is a Child

THEN Drug recommendation is Ampicilin and Drug recommendation is Gentacimin

and Display meningitis Prescription 1

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

Conflict Resolution Methods

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”

Conflict Resolution Methods

Rule 1:

If the forecast is rain [08:16 PM 11/25/2019]

Then the advice is “Take an Umbrella”

Rule 2:

If the weather is wet [10:18 AM 11/26/2019]

Then the advice is “Stay Home”