MODULE 5

Expert systems

- Typical expert system examples. Fuzzy logic: Fuzzy variables, Fuzzy sets and fuzzy set operations, Typical examples using fuzzy sets.

Introduction

- An expert system is a computer program that is designed to solve complex problems and to provide decision-making ability like a human expert.

- The first expert system was Dendral , designed to analyze chemical compounds.

- In order to accomplish feats of apparent intelligence, an expert system relies on two components: **a knowledge base** and **an inference engine**

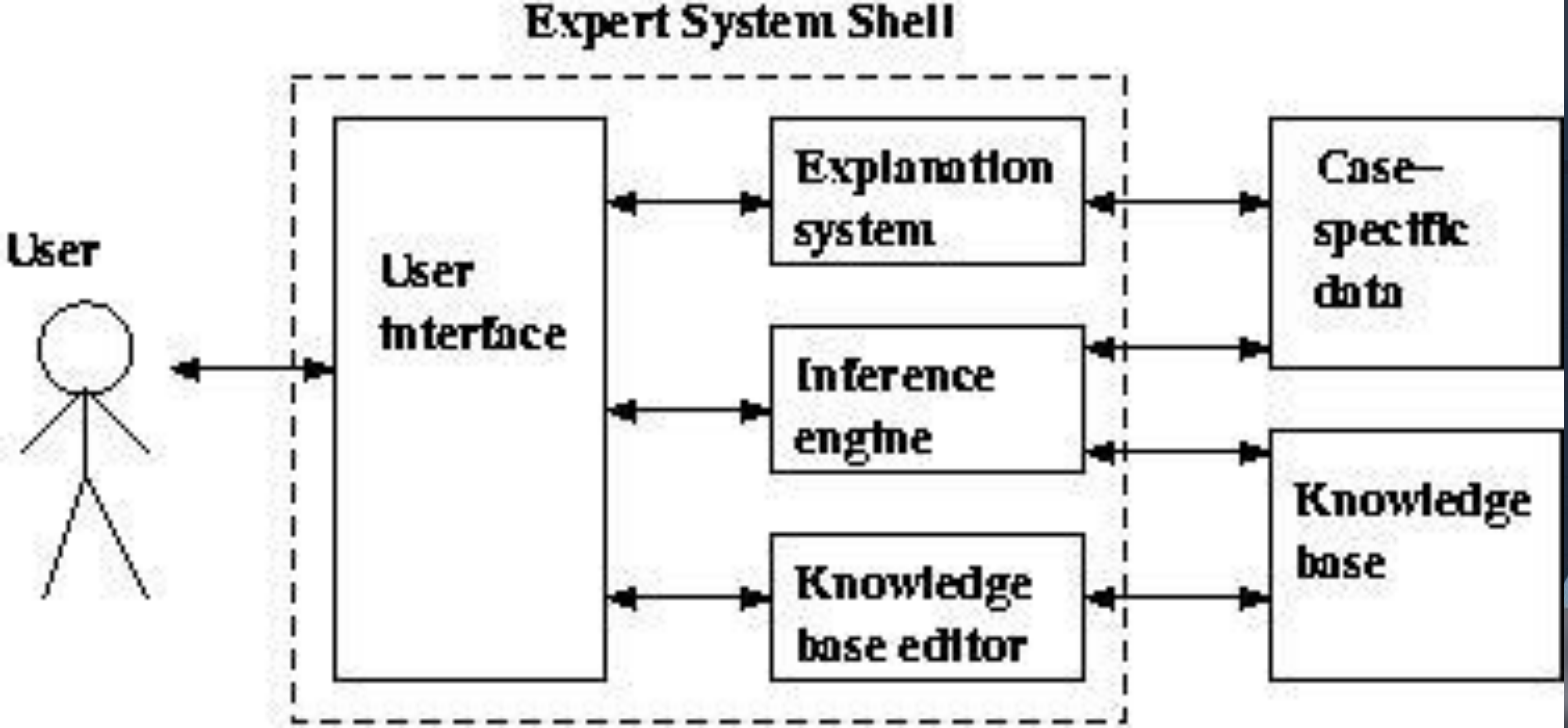
- A knowledge base is an organized collection of facts about the system's domain.

- An inference engine interprets and evaluates the facts in the knowledge base in order to provide an answer.

- Typical tasks for expert systems involve classification, diagnosis, monitoring, design, scheduling, and planning for specialized endeavours.

- Facts for a knowledge base must be acquired from human experts through interviews and observations.
- This knowledge is then usually represented in the form of “if-then” rules.
- The knowledge base of a major expert system includes thousands of rules

Architecture

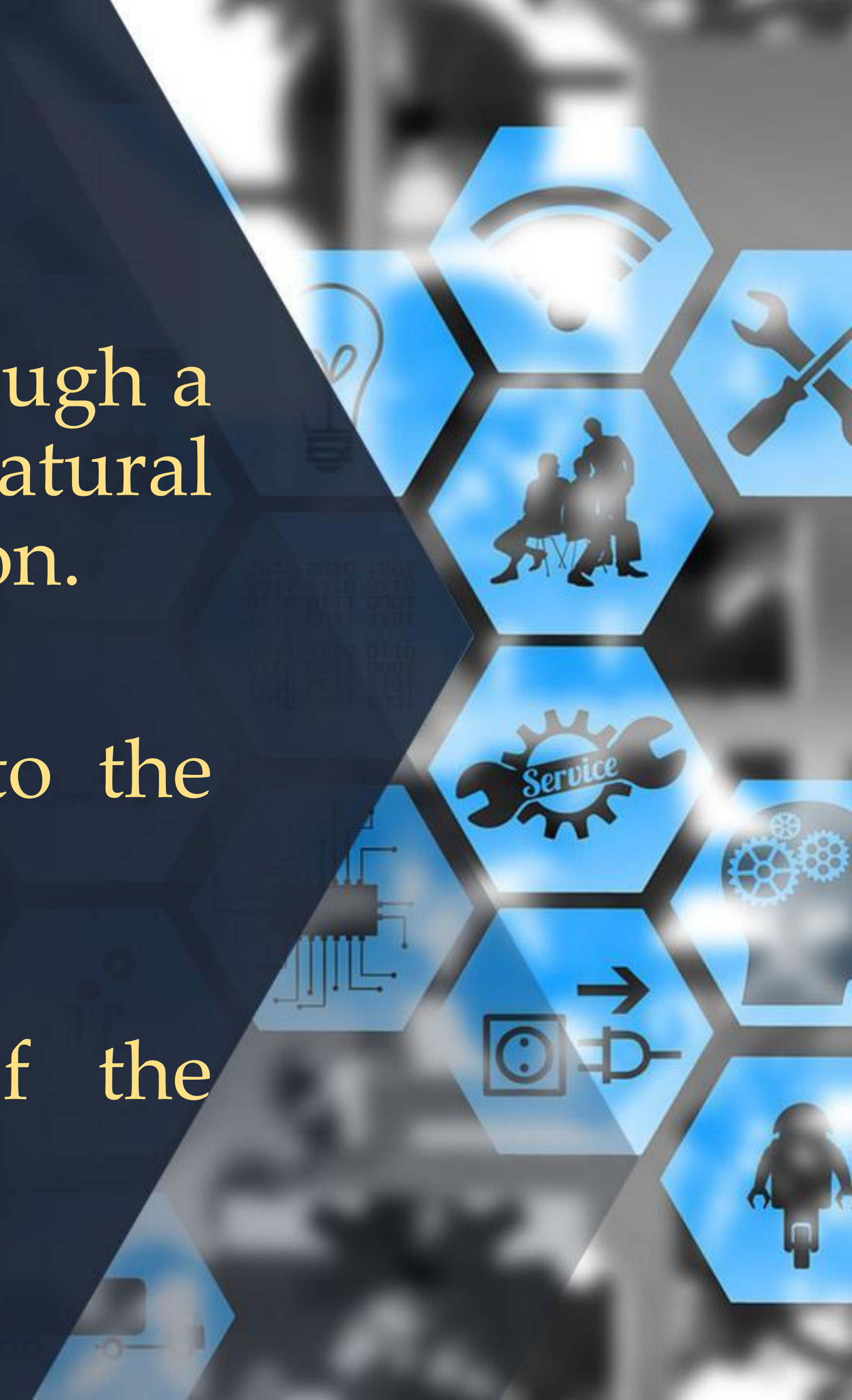


1. User interface

- The user interacts with the system through a user interface which may use menus, natural language or any other style of interaction.

- The user interface is closely linked to the explanation component.

- This is the dialogue component of the system.



- One side of the dialogue involves the user questioning the system and on the other side the system must be able to question the user to establish the existence of evidence.

- The dialogue component has two functions; determines which question to ask next and keep record of the previous questions.

- The dialogue could be one of three styles:
  - System controlled, where the system drives the dialogue through questioning the user.
  - User controlled, where the user drives the consultation by providing information to the system.
  - Mixed control, where both user and system can direct the consultation.

2. Knowledge base

- The knowledgebase is a type of storage that stores knowledge acquired from the different experts of the particular domain. It is considered as big storage of knowledge.

- The more the knowledge base, the more precise will be the Expert System.

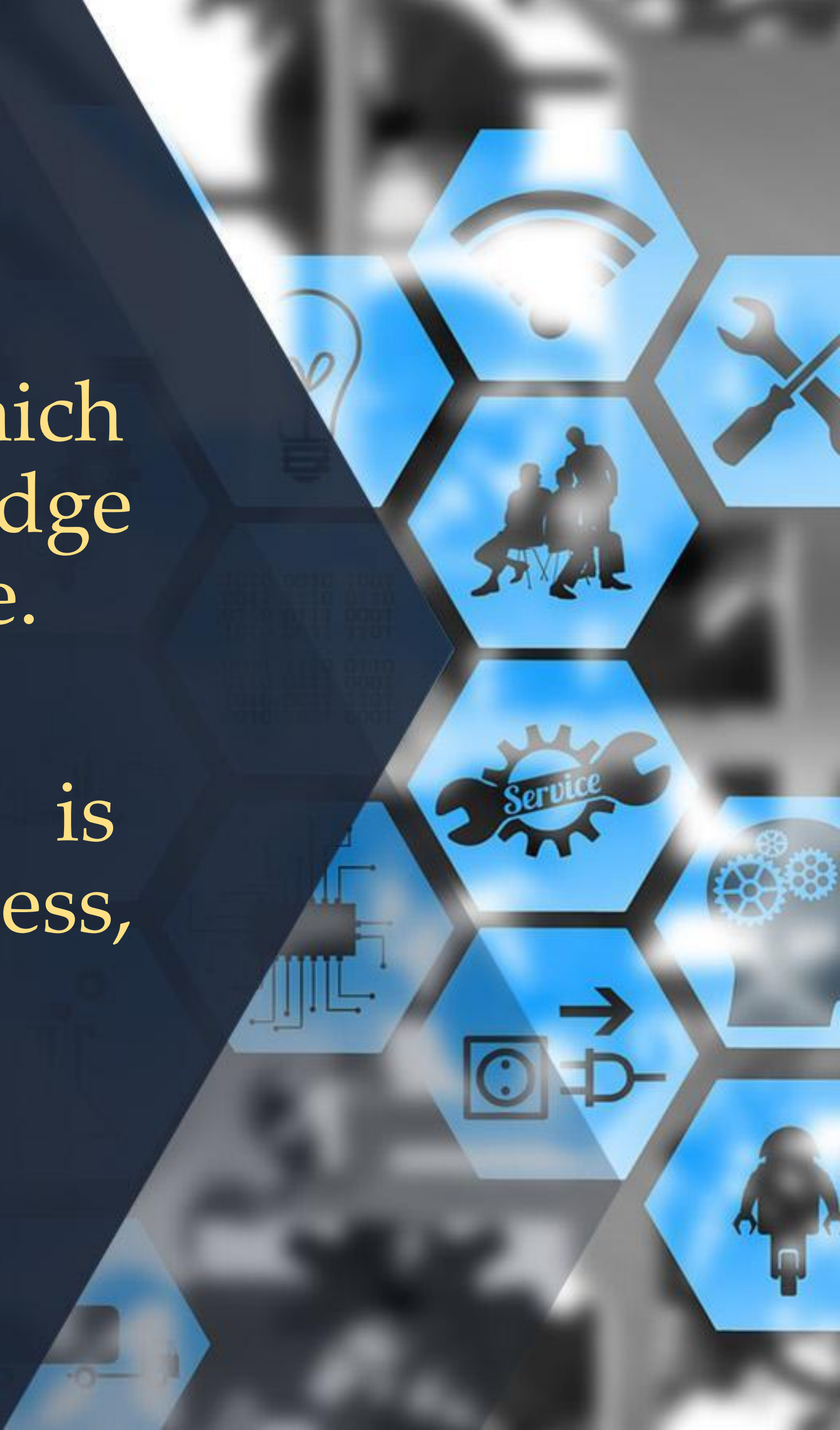
- It is similar to a database that contains information and rules of a particular domain or subject.

- One can also view the knowledge base as collections of objects and their attributes.
- Such as a Lion is an object and its attributes are it is a mammal, it is not a domestic animal, etc
- A knowledge base contains both declarative knowledge (facts about objects, events and situations) and procedural knowledge (information about course of action).

- **Components of Knowledge Base :**

- **Factual Knowledge**: The knowledge which is based on facts and accepted by knowledge engineers comes under factual knowledge.

- **Heuristic Knowledge**: This knowledge is based on practice, the ability to guess, evaluation, and experiences.

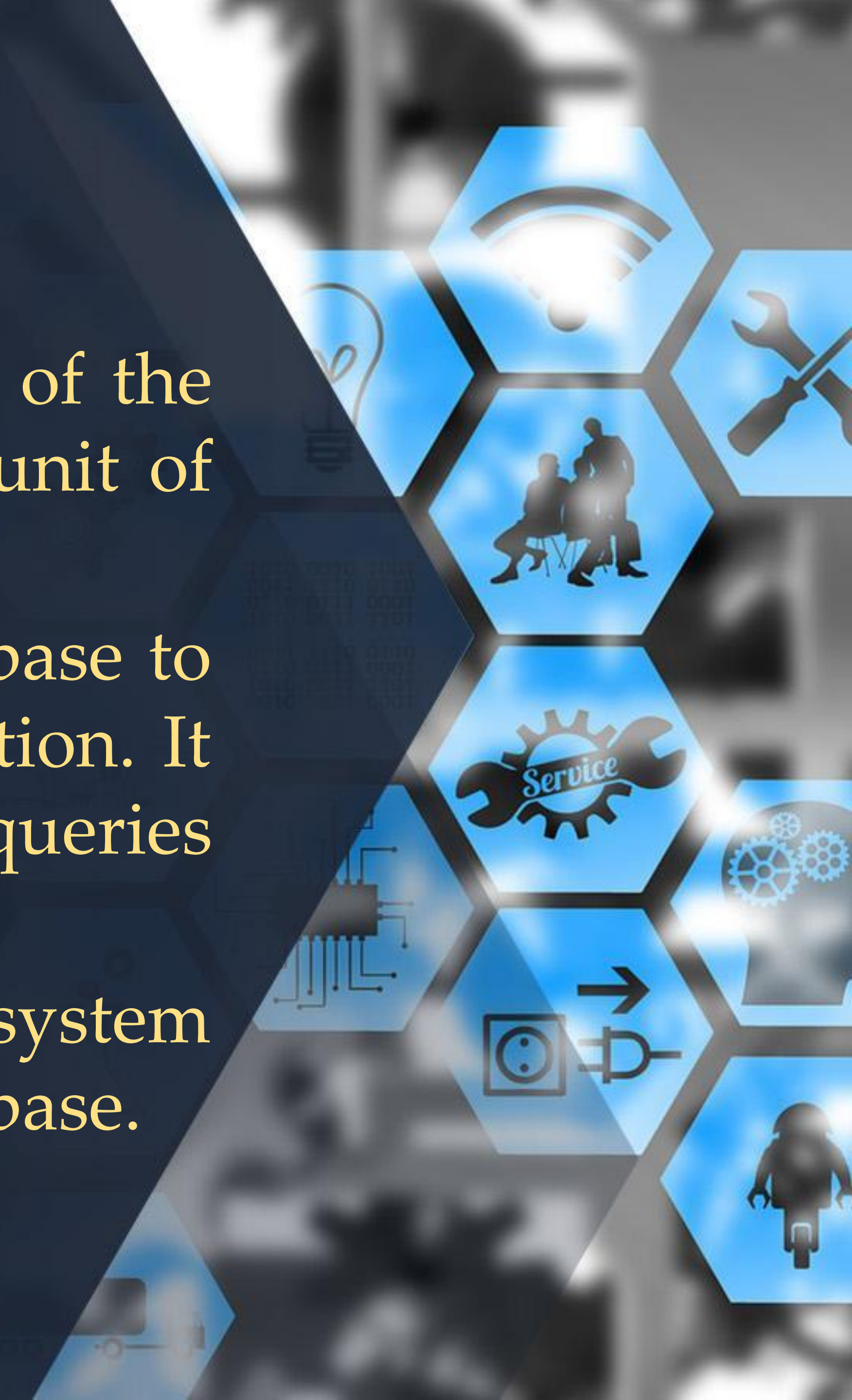


- **Knowledge Representation:** It is used to formalize the knowledge stored in the knowledge base using the If-else rules.

- **Knowledge Acquisitions:** It is the process of extracting, organizing, and structuring the domain knowledge, specifying the rules to acquire the knowledge from various experts, and store that knowledge into the knowledge base

3. Inference engine

- The inference engine is known as the brain of the expert system as it is the main processing unit of the system.
- It applies inference rules to the knowledge base to derive a conclusion or deduce new information. It helps in deriving an error-free solution of queries asked by the user.
- With the help of an inference engine, the system extracts the knowledge from the knowledge base.



- The inference engine uses the information provided to it by the knowledge base and the user to infer new facts.

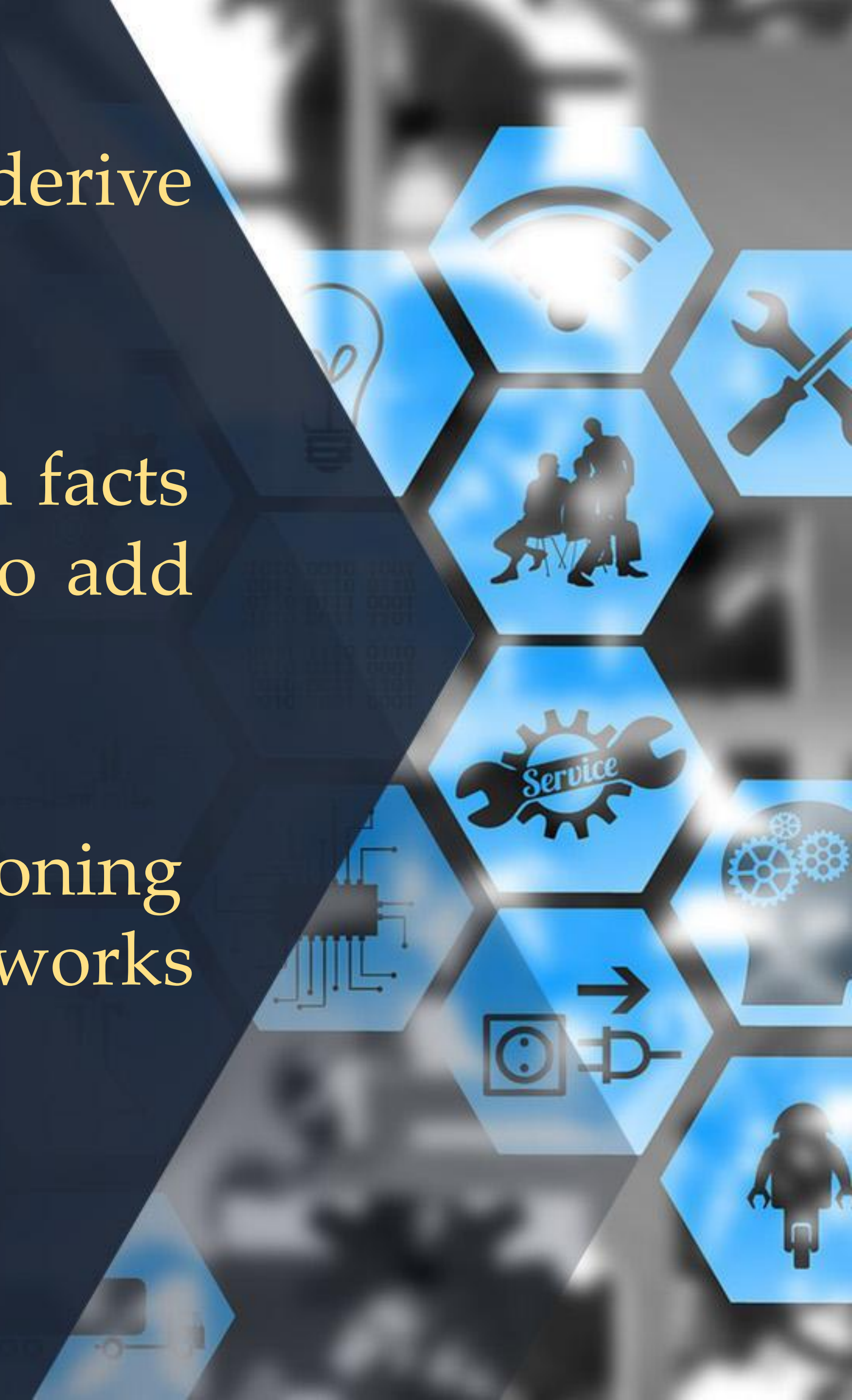
- It also decides which heuristic search techniques are to be used in determining how the rules in the knowledge are to be applied to the problem.

- In fact an inference engine “runs” an expert system, determining which rules are to be invoked, accessing the appropriate rules, executing the rules and determining when an acceptable solution has been found.

- Inference engine uses the below modes to derive the solutions:

- Forward Chaining: It starts from the known facts and rules, and applies the inference rules to add their conclusion to the known facts.

- Backward Chaining: It is a backward reasoning method that starts from the goal and works backward to prove the known facts.



4. Case specific data

- The case specific data includes both data provided by the user and partial conclusions (along with certainty measures) based on this data. In a rule-based system the case specific data will be the elements in working memory

5. Explanation system

- The explanation system allows the program to explain its reasoning to the user.
- It is not acceptable for an expert system to take decisions without being able to provide an explanation for the decisions it has taken.
- Users using these expert systems need to be convinced of the validity of the conclusion drawn before applying it to their domain..

- They also need to be convinced that the solution is appropriate and applicable in their circumstances

6. Knowledge base editor

- Most expert systems provide a mechanism for editing the knowledge base.
- In the simplest case, this is just a standard text editor for modifying rules and data by hand. But many tools include other facilities in their support environment.
- For example, EMYCIN uses automatic book-keeping.

- **Facilities :-**

1. Syntax checking, where the editor uses knowledge about the grammatical structure of the expert system language to help the user input rules with the correct spelling and format.

2. Consistency checking where the system checks the semantics or meanings of the rules and data being entered to see if they conflict with existing knowledge in the system.

3. Knowledge extraction where the editor helps the user enter new knowledge into the system.

7. Expert shell

- One important feature of expert systems is the way they separate domain specific knowledge from more general purpose reasoning and representation techniques.

- The general purpose bit is referred to as an expert system shell.

- The shell provides the inference engine, the user interface, an explanation system and sometimes a knowledge base editor.

- There are numerous commercial expert system shells, each one appropriate for a slightly different range of problems


Service

Roles of individuals who interact with expert systems

- **Domain expert :**

The domain experts are the individuals who are currently experts in solving the problem which the system is intended to solve.

- **Knowledge engineer :**

The knowledge engineer is the person who encodes the expert's knowledge in a declarative form that can be used by the expert system.

- **User :**

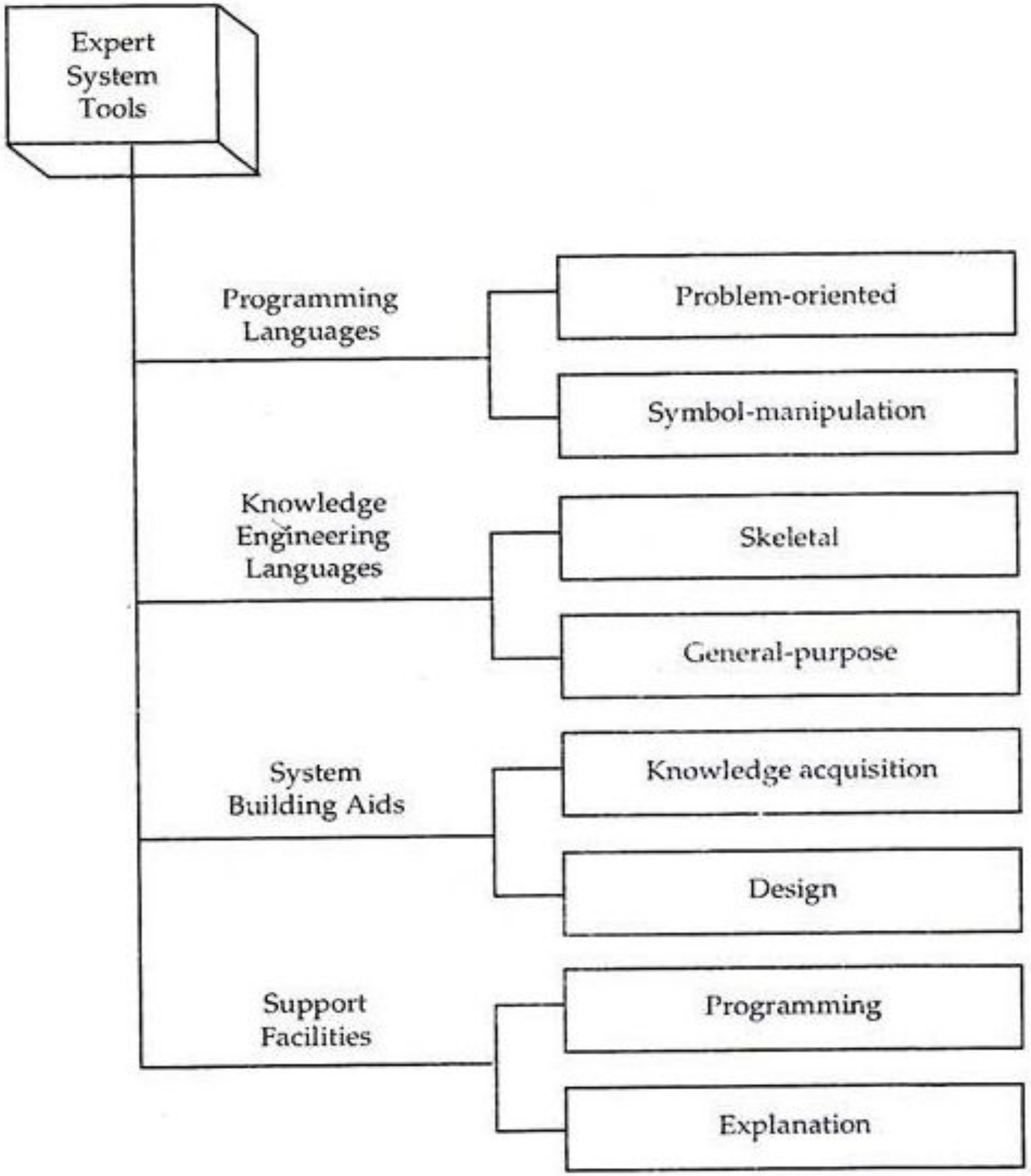
The user is the person who will be consulting with the system to get advice which would have been provided by the expert.

- **System engineer :**

The system engineer builds the user interface, designs the declarative format of the knowledge base, and implements the inference engine.

Languages and tools

- Expert system tools are programming systems which simplify the job of constructing an expert system.
- They range from very high-level programming language to low-level support facilities.



Expert System Tools
Programming Languages
Problem-oriented
Symbol-manipulation
Knowledge Engineering Languages
Skeletal
General-purpose
System Building Aids
Knowledge acquisition
Design
Support Facilities
Programming
Explanation

- **Programming languages** :
- Most important programming languages used for expert system applications are generally either problem-oriented languages such as FORTRAN and PASCAL, or symbol-manipulation such as LISP and PROLOG.

- Symbol-manipulation languages are designed for artificial intelligence applications; for example, LISP has mechanism for manipulating symbols in the form of list structures.

- A list is simply a collection of items enclosed by parenthesis, where each item can be either a symbol or another list.

- List structures are useful building blocks for representing complex concepts.

- Expert systems are also written in CLIPS (C Language Integrated Production System developed in mid 1980s).

- The major advantages of these languages as compared to conventional programming languages, is the simplicity of the addition, elimination or substitution of new rules and memory management capabilities

- **Knowledge engineering languages :**
- A knowledge engineering language is a sophisticated tool for developing expert systems.
- It consists of an expert system building language integrated into an extensive support environment.
- A knowledge engineering language is a type of programming language designed to construct and debug expert system.

- Knowledge engineering languages can be categorized as either skeletal systems or general-purpose systems.
- A skeletal knowledge engineering language is a stripped down expert system, that is, an expert system with its domain-specific knowledge removed leaving only the inference engine and support facilities.
- The general-purpose knowledge engineering language can handle many different problem areas such as knowledge extraction, giving inference or making user interface though its use is rather tedious

- **System-building aids**
- The system-building aids consist of programs which help acquire and represent the domain expert's knowledge and programs which help design the expert system under construction.
- These programs address very difficult tasks; many are research tools just beginning to evolve into practical and useful aids, although a few are offered as full-blown commercial systems.

- Those which exist fall into two major categories; design aids and knowledge acquisition aids.
- The AGE system exemplifies design aids
- TEIRSIAS, MOLE and SALT exemplify knowledge acquisition, TIMM system construction
- SEEK knowledge refinement aids

- **Tool support environment facilities :**
- These are software packages which come with each tool to make it more user friendly and more efficient.

Typical expert systems

1. Dendral

- Dendral was a project in artificial intelligence of the 1960s.
- Its primary aim was to study hypothesis formation and discovery in science.
- For that, a specific task in science was chosen: help organic chemists in identifying unknown organic molecules, by analyzing their mass spectra and using knowledge of chemistry.
- The first expert system because it automated the decision-making process and problem-solving behavior of organic chemists.

2. MYCIN

- MYCIN was an early backward chaining expert system that used artificial intelligence to identify bacteria causing severe infections, such as bacteremia and meningitis, and to recommend antibiotics, with the dosage adjusted for patient’s body weight.
- The Mycin system was also used for the diagnosis of blood clotting diseases.

3. DART (Diagnostic Assistance Reference Tool)

- DART is a joint project of the Stanford University and IBM that explores the application of artificial intelligence techniques to the diagnosis of computer faults.
- It assists a technician in finding the faults in a computer system.
- The primary goal of the DART Project was to develop programs that capture the special design knowledge and diagnostic abilities of the experts and to make them available to field engineers.

4. XCON (eXpert CONfigure)

- System with the ability to select specific software to generate a computer system as per user preference, written in 1978.

5. PXDES

- This system could easily determine the type and the degree of lung cancer in patients based on limited data.

6. CaDet

- This is a clinical support system that identifies cancer in early stages.

7. PIP (Process Invention Procedure)

- It is an hierarchical expert system for the synthesis of chemical process flowsheets.
- It uses a combination of qualitative knowledge, that is, heuristics and quantitative knowledge, that is, design and cost calculations, arranged in a hierarchic manner.

8. INTERNIST

- INTERNIST-I was designed between 1972 and 1973 to provide computer assisted diagnosis in general internal medicine by attempting to model the reasoning of clinicians

Benefits

- Improves decision-making quality.
- Cost-effective, as it trims down the expense of consulting human experts when solving a problem.
- Provides fast and robust solutions to complex problems in a specific domain.
- It gathers scarce knowledge and uses it efficiently.
- Offers consistency when providing answers for repetitive issues.
- Maintains a good amount of information.
- Provides fast and accurate answers.
- Provides a proper explanation for decision making.
- Solves complex and challenging issues.
- Works steadily without fatigue

Limitations

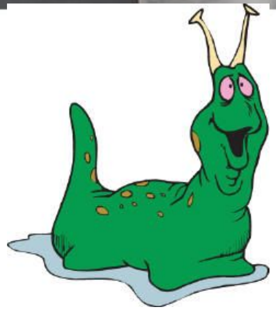
- Not capable of making decisions in extraordinary situations.
- Garbage-in Garbage-out (GIGO), if there is an error in the knowledge base, we are bound to get wrong decisions.
- The maintenance cost is more.
- Each problem is different, and expert systems have some limitations when it comes to solving varied problems. In such cases, a human expert is more creative.

Fuzzy logic

WHAT IS FUZZY LOGIC?

- Definition of fuzzy
  - Fuzzy – "not clear, distinct, or precise; blurred"
- Definition of fuzzy logic
  - A form of knowledge representation suitable for notions that cannot be defined precisely, but which depend upon their contexts.

TRADITIONAL REPRESENTATION OF LOGIC



Slow

Speed = 0

Fast

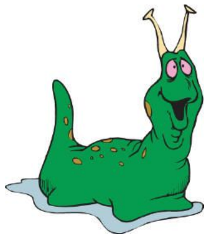
Speed = 1

```
bool speed;
get the speed
if ( speed == 0) {
    //  speed is slow
}
else {
    //  speed is fast
}
```

FUZZY LOGIC REPRESENTATION

- For every problem must represent in terms of fuzzy sets.

- What are fuzzy sets?



Slowest

[0.0 – 0.25]

Slow

[0.25 – 0.50]

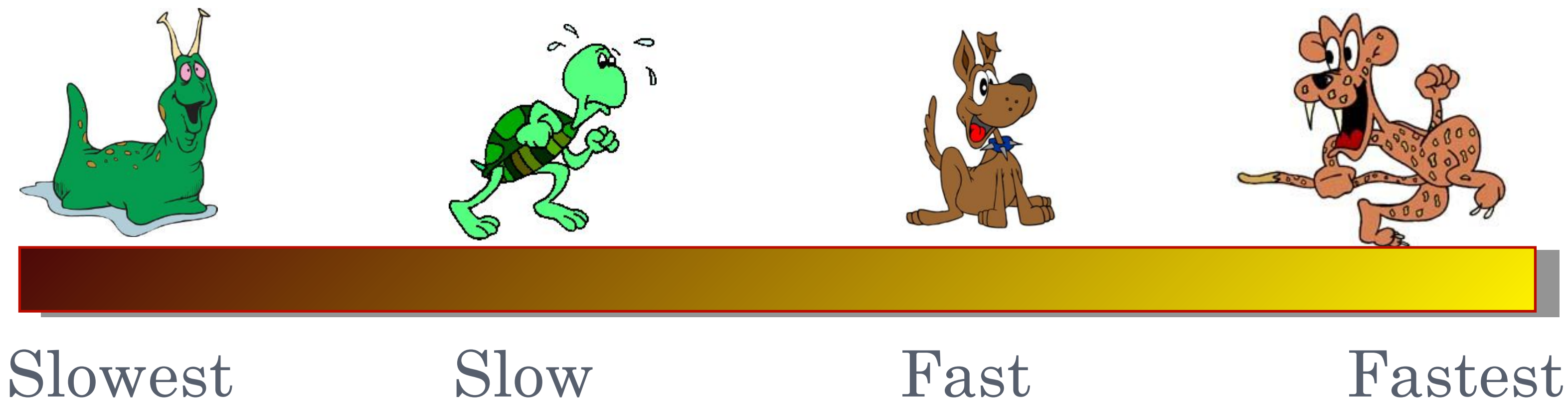
Fast

[0.50 – 0.75]

Fastest

[0.75 – 1.00]

FUZZY LOGIC REPRESENTATION CONT.




```
float speed;
get the speed
if ((speed >= 0.0)&&(speed < 0.25)) {
    //  speed is slowest
}
else if ((speed >= 0.25)&&(speed < 0.5
{
    //  speed is slow
}
else if ((speed >= 0.5)&&(speed < 0.75
{
    //  speed is fast
}
else // speed >= 0.75 && speed < 1.0
{
    //  speed is fastest
}
```

Fuzzy Logic

- Fuzzy Logic tries to capture the human ability of reasoning with imprecise information.

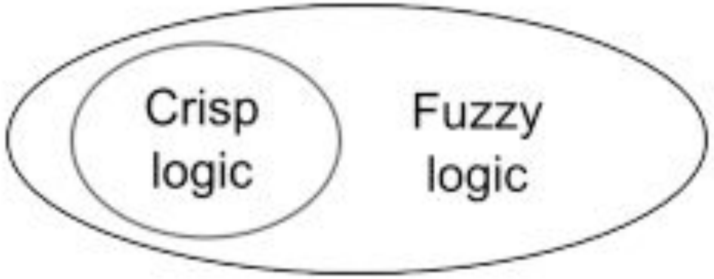
- Fuzzy Logic is a form of multi-valued logic derived from fuzzy set theory to deal with reasoning that is approximate rather than precise.

Fuzzy Logic

- Fuzzy logic is not a vague logic system, but a system of logic for dealing with vague concepts.

- As in fuzzy set theory the set membership values can range (inclusively) between 0 and 1, in fuzzy logic the degree of truth of a statement can range between 0 and 1 and is not constrained to the two truth values true/false as in classic predicate logic.

- Fuzzy logic is based on the theory of fuzzy sets, which is a generalization of the classical set theory.
- Saying that the theory of fuzzy sets is a generalization of the classical set theory means that the latter is a special case of fuzzy sets theory.
- To make a metaphor in set theory speaking, the classical set theory is a subset of the theory of fuzzy sets, as



Fuzzy sets

- Fuzziness or vagueness can be found in many areas of daily life.
- This is more frequently seen in which human judgment, evaluation, and decisions are important. The reason for this is that our daily communication uses "natural languages" and a good part of our thinking is done in it.
- In these natural languages the meaning of words is very often vague. Examples are words such as "birds" (how about penguins, bats, etc.?), "red roses", but also terms such as "tall men", "beautiful women", etc.
- The term "tall men" is fuzzy because the meaning of "tall" is fuzzy and dependent on the context (height of observer, culture, etc.).

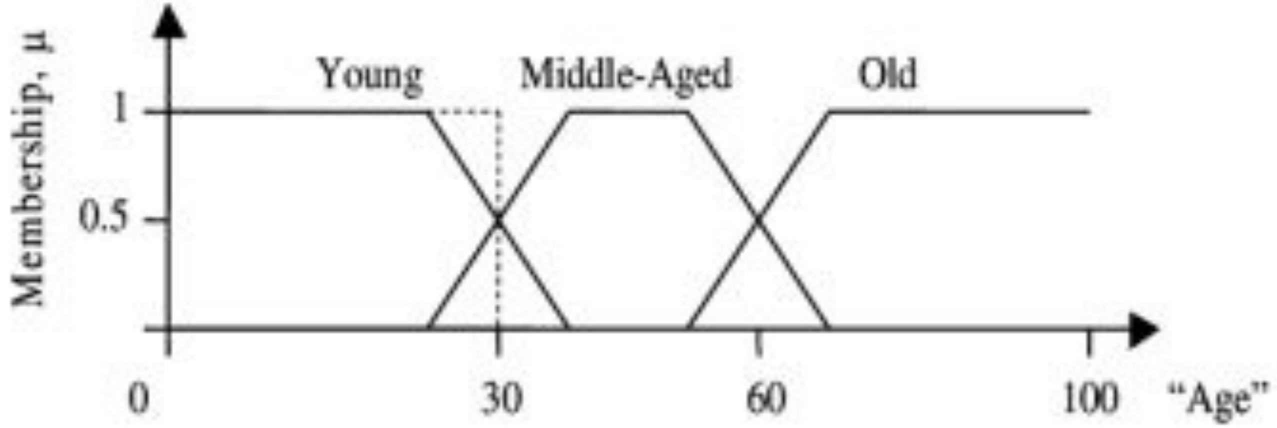
Example

- Imagine the set of all young men residing in a small town. Since the number of people residing in the town is not very large it is indeed possible to prepare a list of all persons in the town. But can we prepare a list of all young men in the town? It is not possible because the attribute of being “young” is not well defined, is vague and is imprecise. We say that he attribute is a “fuzzy” attribute

- To describe the set of young men in the town, we may arbitrarily assign a number to each person which may indicate the “youngness” of the person.
- We assign the smallest number to a person having the least “youngness” property and the highest value to those who have the highest “youngness” property.
- The assignment of numbers is purely arbitrary.

- However it should reflect a human being’s understanding the notion of “youngness”: a higher value must indicate a higher level of “youngness”.

- A set to whose elements are assigned numbers like this is called a **fuzzy set**



Crisp sets

- The sets studied in classical set theory as “crisp sets”.
- In the case of crisp sets, given an object, we can say unambiguously whether the object is in the crisp set or not.
- In the case of crisp sets also, we can assign numbers to objects: Assign 1 if the object is in the set and assign 0 if it is not in the set.
- In this sense, crisp sets, that is sets considered in elementary set theory can also be regarded as special fuzzy sets

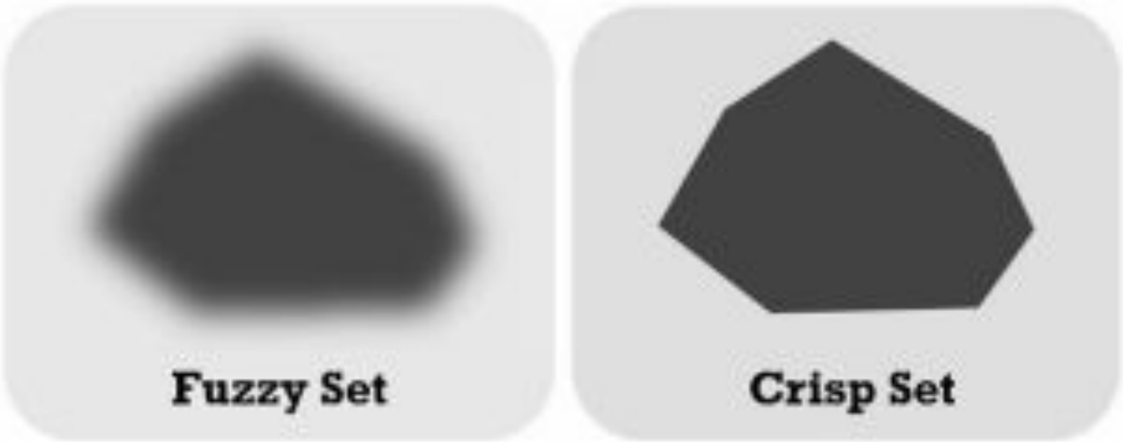


Figure 11.2: Fuzzy set vs. crisp set

Fuzzy sets- Definition

A **fuzzy set** is a pair (U, m) where U is a set (assumed to be non-empty) and $m : U \to [0, 1]$ a function. The set U is called *universe of discourse*. The function m is called the *membership function*. For each $x \in U$, the value $m(x)$ is called the *grade of membership* of x in (U, m).

The fuzzy set (U, m) is often denoted by a single letter A. In such a case, the membership function is denoted by μ_A.

Types of membership

Let $x \in U$. Then x is said to be

- *not included* in the fuzzy set (U, m) if $m(x) = 0$ (no member),
- *fully included* if $m(x) = 1$ (full member),
- *partially included* if $0 < m(x) < 1$ (fuzzy member).

- Support, kernel, α-cut

Let $A = (U, m)$ be a fuzzy set. The support of A is the crisp set

$$\text{Supp}\,(A) = \{x \in U : m(x) > 0\}.$$

The kernel of A is the crisp set

$$\text{Kern}\,(A) = \{x \in : m(x) = 1\}.$$

The α-cut (also called the α-level cut of A), denoted by $A^{\geq \alpha}$ or A_α, is defined as the crisp set

$$A^{\geq \alpha} = \{x \in U : \mu_A(x) \leq \alpha\}.$$

Let $A = (U, m)$ be a fuzzy set and let U be a finite set. Let the support of A be Supp $(U) = \{x_1, x_2, \ldots, x_n\}$. Then the fuzzy set A is sometimes written as a set of ordered pairs as follows:

$$A = \{(x_1, m(x_1)), (x_2, m(x_2)), \cdots, (x_n, m(x_n))\}$$

The following notation

$$A = \{m(x_1)/x_1, m(x_2)/x_2, \ldots, m(x_n)/x_n\}$$

as well as the notation

$$A = m(x_1)/x_1 + m(x_2)/x_2 + \ldots + m(x_n)/x_n$$

are also used to denote the fuzzy set A. Still others write

$$A = \left\{ \frac{m(x_1)}{x_1}, \frac{m(x_2)}{x_2}, \ldots, \frac{m(x_n)}{x_n} \right\}.$$

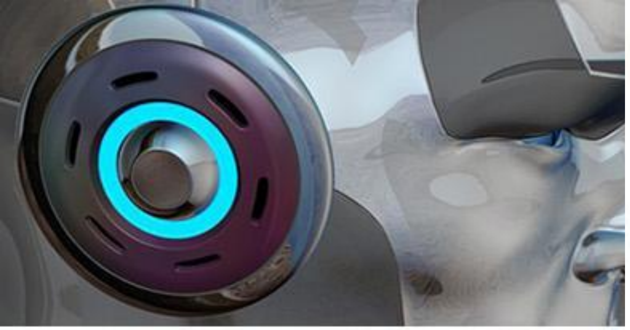
Note that in these representations, the symbols "/" and "+" are just symbols and they do not represent the corresponding arithmetical operations.

- Crisp set as a fuzzy set

Let A be a crisp subset of the universe of discourse U. A can be treated as a fuzzy set $A = (U, m)$ where the membership function m is defined as

$$m(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{if } x \notin A \end{cases}$$

The membership function of a crisp set is called the characteristic function of the set.



- Example

Consider the set

$$U = \{5, 10, 15, 20, 30, 35, 40, 45, 60, 70\}$$

and the function $m : U \to [0, 1]$ defined by the following table:

x	5	10	15	20	30	35	40	45	60	70
$m(x)$	1	1	1	1	0.6	0.5	0.4	0.2	0	0

$A = (U, m)$ is a fuzzy set. The support of A is

$$\text{Supp}\,(A) = \{5, 10, 15, 20, 30, 35, 40, 45\}$$

and the kernel of A is

$$\text{Kern}\,(A) = \{5, 10, 15, 20\}.$$

The fuzzy set may be represented as

$$A = \{1/5, 1/10, 1/15, 1/20, 0.6/30, 0.5/35, 0.4/40, 0.2/45\}$$

or as

$$A = 1/5 + 1/10 + 1/15 + 1/20 + 0.6/30 + 0.5/35 + 0.4/40 + 0.2/45.$$

Example 2

A realtor wants to classify the houses he offers to his clients. One indicator of comfort of these houses is the number of bedrooms in it. Let

$$U = \{1, 2, 3, 4, ..., 10\}$$

be the set of available types of houses described by $x =$ number of bedrooms in a house. Then the fuzzy set "comfortable type of house for a 4-person family" may be described as

$$A = \{0.2/1, 0.5/2, 0.8/3, 1.0/4\ 0.7/5, 0.3/6)\}.$$

Example 3

The fuzzy set defined as “integers close to 10” may be specified by

$$A = 0.1/7 + 0.5/8 + 0.8/9 + 1/10 + 0.8/11 + 0.5/12 + 0.1/13$$

Set-theoretic operations for fuzzy sets

Definitions

Let A and B be two fuzzy sets in a universe U.

1. Equality

A and B are said to be equal, denoted by A = B, if $\mu A(x) = \mu B(x)$ for all x in U.

2. Union

The union of A and B, denoted by $A \cup B$, is the fuzzy set C whose membership function is defined by,

$$\mu C(x) = \max\{\mu A(x), \mu B(x)\}, x \in U.$$

3. Intersection

The intersection of A and B, denoted by A∩B, is the fuzzy set D whose membership function is defined by

$$\mu D(x) = \min\{\mu A(x), \mu B(x)\}, x \in U.$$

4. Complement

The complement of A, denoted by A, is the fuzzy set E whose membership function is defined by

$$\mu E(x) = 1 - \mu A(x), x \in U$$

1. The definitions of the membership functions of the union, intersection and complement of fuzzy sets may be given in the following forms also:

- $\mu A \cup B(x) = \max\{\mu A(x), \mu B(x)\}$
- $\mu A \cap B(x) = \min\{\mu A(x), \mu B(x)\}$
- $\mu A(x) = 1 - \mu A(x)$

2. The support of a fuzzy set is a crisp set. We have

$$\mathrm{Supp}\,(A \cup B) = \mathrm{Supp}\,(A) \cup \mathrm{Supp}\,(B)$$

$$\mathrm{Supp}\,(A \cap B) = \mathrm{Supp}\,(A) \cap \mathrm{Supp}\,(B)$$

$$\mathrm{Supp}\,(\overline{A}) = \overline{\mathrm{Supp}\,(A)}$$

3. If A is a fuzzy set in a universe U then in general we have $A \cup \overline{A} \neq U, \qquad A \cap \overline{A} \neq \varnothing.$

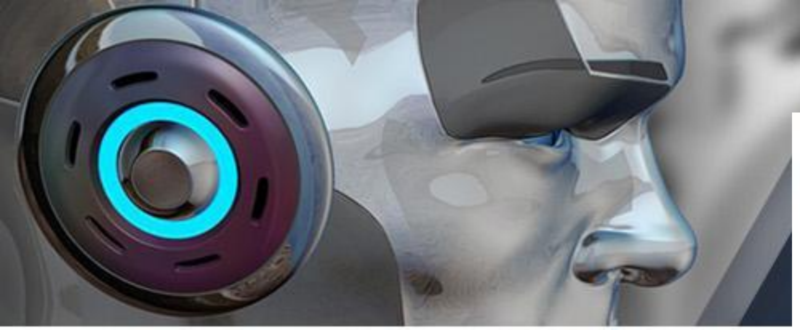
However, equalities hold if A is a crisp set considered as a fuzzy set

Example 1

Given the fuzzy sets

$$A = \{0.3/2, 0.4/3, 0.1/4, 0.8/5, 1.0/6\}$$
$$B = \{0.7/4, 0.5/5, 1.0/6, 0.02/7, 0.75/8\}$$

find $A \cup B$ and $A \cap B$.



- Solution

We have:

$$\text{Supp}(A) = \{2, 3, 4, 5, 6\}$$
$$\text{Supp}(B) = \{4, 5, 6, 7, 8\}$$

Let us compute $A \cup B$.

$$\begin{aligned}\text{Supp}(A \cup B) &= \text{Supp}(A) \cup \text{Supp}(B) \\ &= \{2, 3, 4, 5, 6, 7, 8\}\end{aligned}$$

We have

$$\mu_{A \cup B}(x) = \max\{\mu_A(x), \mu_B(x)\}.$$

Hence

$$
\begin{aligned}
\mu_{A\cup B}(2) &= \max\{\mu_A(2), \mu_B(2)\} \\
&= \max\{0.3, 0\} \quad (2 \notin \text{Supp}\,(B)) \\
&= 0.3 \\
\mu_{A\cup B}(3) &= \max\{\mu_A(3), \mu_B(3)\} \\
&= \max\{0.4, 0\} \quad (3 \notin \text{Supp}\,(B)) \\
&= 0.4 \\
\mu_{A\cup B}(4) &= \max\{\mu_A(4), \mu_B(4)\} \\
&= \max\{0.1, 0.7\} \\
&= 0.7 \\
\cdots &= \cdots \\
\mu_{A\cup B}(8) &= \max\{\mu_A(8), \mu_B(8)\} \\
&= \max\{0, 0.75\} \quad (8 \notin \text{Supp}\,(A)) \\
&= 0.75
\end{aligned}
$$

Therefore

$$A \cup B = \{0.3/2, 0.4/3, 0.7/4, 0.8/5, 1.0/6, 0.02/7, 0.75/8\}.$$

Now, let us calculate $A \cap B$.

$$\begin{aligned}\text{Supp}\,(A \cap B) &= \text{Supp}\,(A) \cap \text{Supp}\,(B)\\ &= \{4, 5, 6\}\end{aligned}$$

We have

$$\mu_{A\cap B}(x) = \min\{\mu_A(x), \mu_B(x)\}.$$

Hence

$$\begin{aligned}\mu_{A\cap B}(4) &= \min\{\mu_A(4), \mu_B(4)\}\\ &= \min\{0.1, 0.7\}\\ &= 0.1\\ \mu_{A\cap B}(5) &= \min\{\mu_A(5), \mu_B(5)\}\\ &= \max\{0.8, 0.5\}\\ &= 0.5\\ \mu_{A\cap B}(6) &= \min\{\mu_A(6), \mu_B(6)\}\\ &= \max\{1, 0, 1, 0\}\\ &= 1.0\end{aligned}$$

Therefore

$$A \cap B = \{0.1/4, 0.5/5, 1.0/6\}$$

Example 2

Let the universe be $U = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ and let

$$A = \{0.2/1, 0.4/2, 0.6/3, 0.8/4, 1.0/5\}.$$

Compute $\bar{A}$.

Solution

By definition

$$\mu_{\overline{A}}(x) = 1 - \mu_A(x), \text{ for all } x \in U.$$

Therefore, we have:

$$\overline{A} = \{0.8/1, 0.6/2, 0.4/3, 0.2/4, 1.0/6, 1.0/7, 1.0/8, 1.0/9, 1.0/10\}.$$

1. Fuzzy empty set

   A fuzzy set A is said to be empty if $\mu A(x) = 0$ for all $x \in U$, that is, if Supp (A) = $\varnothing$.

2. Fuzzy subset

   A fuzzy set A is said to be a subset of a fuzzy set B, denoted by $A \subseteq B$, if $\mu A(x) \leq \mu B(x)$ for all $x \in U$.

3. Disjoint fuzzy sets

   Two fuzzy sets A and B are said to be disjoint if $\mu A(x) = 0$ or $\mu B(x) = 0$ for all $x \in U$, that is, if Supp $(A \cap B) = \varnothing$.

Example :

1. Let

$$A = \{0.12/2, 0.23/4\}, \quad B = \{0.15/2, 0.37/4, 0.15/6\},$$

Then $A \subseteq B$.

2. Let $U = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ and let

$$A = \{0.1/1, 0.2/2, 0.3/3\}, \quad B = \{0.2/6, 0.4/7, 0.6/8\}.$$

Then

$$\begin{aligned} \text{Supp}\,(A) &= \{1, 2, 3\} \\ \text{Supp}\,(B) &= \{6, 7, 8\} \\ \text{Supp}\,(A \cap B) &= \text{Supp}\,(A) \cap \text{Supp}\,(B) \\ &= \varnothing \end{aligned}$$

Hence A and B are disjoint fuzzy sets.

Properties of set operations on fuzzy sets

1. **Commutativity**

$$A \cup B = B \cup A$$

$$A \cap B = B \cap A.$$

2. **Associativity**

$$A \cup (B \cup C) = (A \cup B) \cup C)$$

$$A \cap (B \cap C) = (A \cap B) \cap C$$

3. **Distributivity**

$$A \cup (B \cap C) = (A \cup B) \cap (A \cup C),$$

$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$$

4. **Identity**

$$A \cup \varnothing = A,\ A \cup X = A$$

$$A \cap \varnothing = \varnothing,\ A \cap X = A$$

5. **Involution**

$$\overline{(\overline{A})} = A.$$

6. **De Morgan's laws**

$$\overline{A \cup B} = \overline{A} \cap \overline{B}$$

$$\overline{A \cap B} = \overline{A} \cup \overline{B}.$$

Example

Given the universe $U = \{a, b, c, d, e, f, g, h\}$ and the fuzzy sets

$$A = \{0.1/a, 0.2/b, 0.3/c, 1.0/d, 0.1/e, 0.5/h\}$$
$$B = \{0.75/b, 0.2/c, 0.35/d, 0.4/e, 0.5/f\}$$
$$C = \{1.0/e, 0.9/f, 0.8/g, 0.7/h\}$$

verify the distributivity properties.

Solution

$$B \cup C = \{0.75/b, 0.2/c, 0.35/d, 1.0/e, 0.9/f, 0.8/g, 0.7/h\}$$
$$A \cap (B \cup C) = \{0.2/b, 0.2/c, 0.35/d, 0.1/e, 0.5/h\}$$
$$A \cap B = \{0.2/b, 0.2/c, 0.35/d, 0.1/e\}$$
$$A \cap C = \{0.1/e, 0.5/h\}$$
$$(A \cap B) \cup (A \cap C) = \{0.2/b, 0.2/c, 0.35/d, 0.1/e, 0.5/h\}$$

Hence we have

$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C).$$

In a similar way, the other distributivity property can be verified.

Fuzzy variables

- An ordinary variable, or a crisp variable, is a variable whose possible values are elements of a given set. For example, a real variable is a variable whose possible values are elements of the set of real numbers.

Definition :-

- A fuzzy variable x is a variable whose possible values are fuzzy sets in some universe of discourse.

Example 1

We define a fuzzy variable x as follows. Let the universe of discourse be

$$U = \{1, 2, 3, 4, 5\}.$$

The possible values of x are fuzzy sets in U. A possible value of x is

$$A = \{0.1/1, 0.7/2, 0.8/3, 0.2/4, 0/5\},$$

or

$$B = \{1.0/1, 0.8/2, 0.6/3, 0.4/4, 0.2/5\}.$$

Example 2

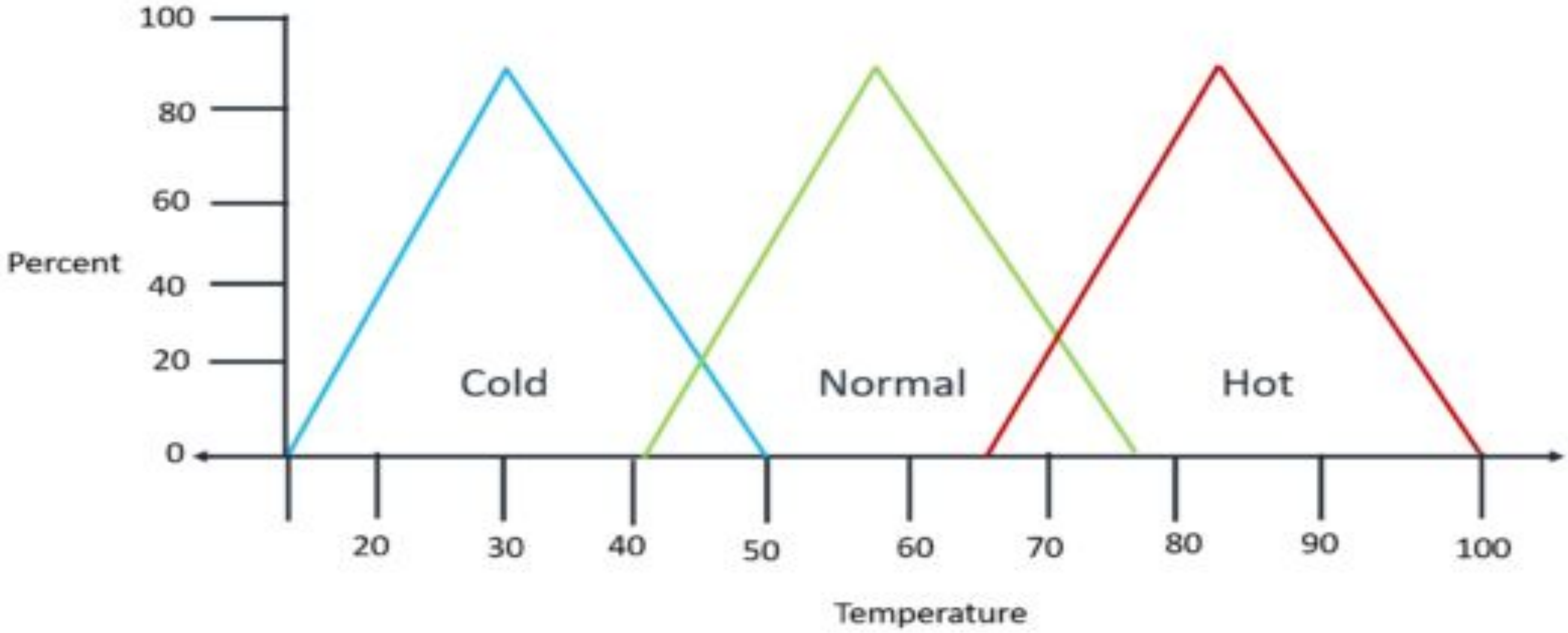
- Consider a variable which denotes the age of a person in years. We may denote this variable by age.
- As a crisp variable, age can assume nonnegative integers as a values.
- We may take age as a fuzzy variable. To do this, we have to define a universe of discourse U which we may take as the set of all nonnegative integers.

- As a fuzzy variable the possible values can that can be assigned to age are fuzzy sets in U.
- These fuzzy sets may denoted by linguistic terms indicative of the age of a person, such as infant, young, middle-aged, old, etc.
- Thus the fuzzy variable age can be thought of as a variable whose possible values are such linguistic terms.
- However, it is to be emphasised that the value is not the linguistic term but the fuzzy set represented by the term.

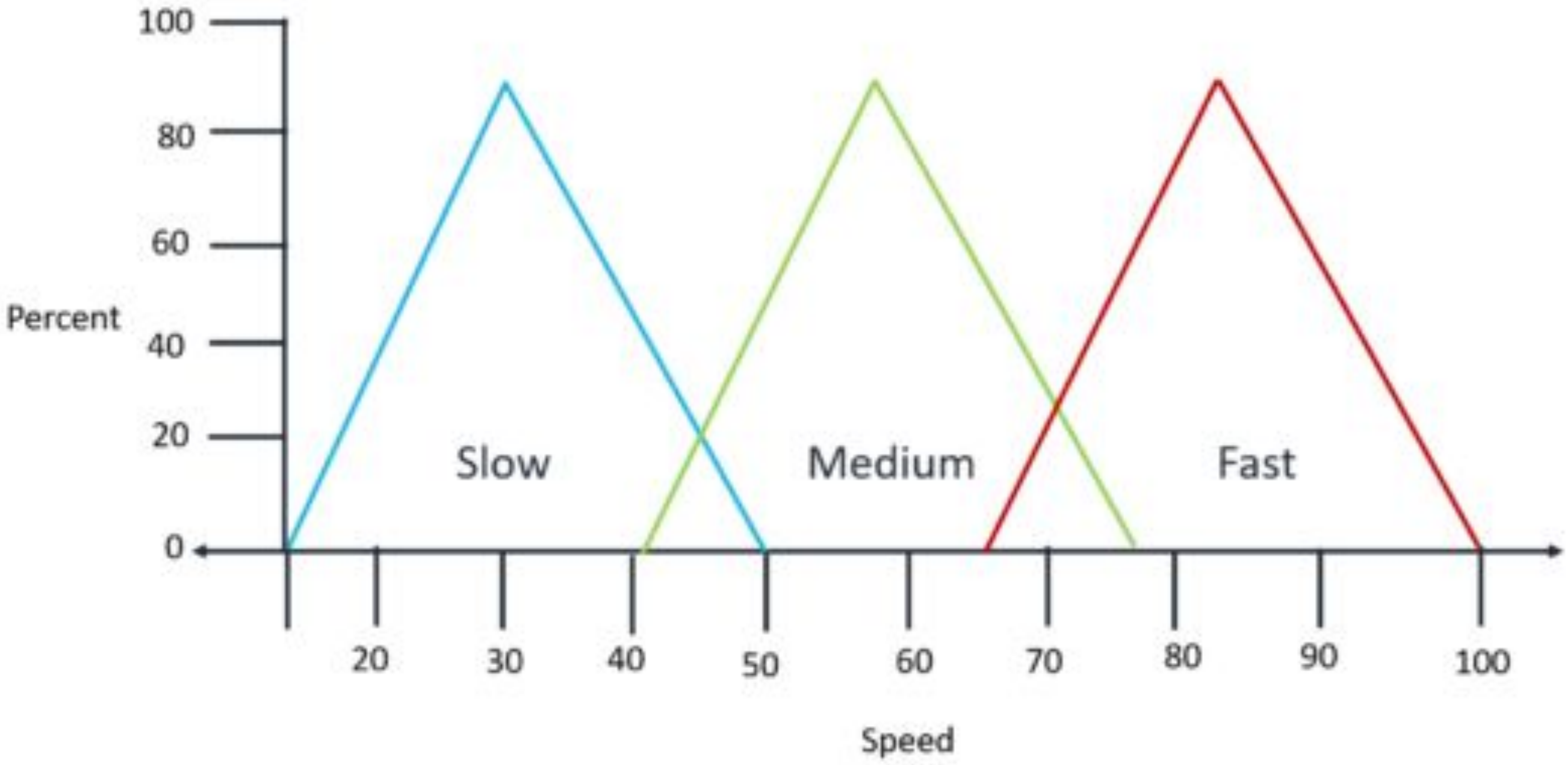
Application example

- The design of a fuzzy logic system starts with a set of membership functions for each input and a set for each output. A set of rules is then applied to the membership functions to yield a crisp output value. Let's take an example of process control and understand fuzzy logic.

- **Step 1**
- Here, **Temperature** is the input and **Fan Speed** is the output. You have to create a set of membership functions for each input. A membership function is simply a graphical representation of the fuzzy variable sets. For this example, we will use three fuzzy sets, **Cold, Warm** and **Hot**. We will then create a membership function for each of three sets of temperature:

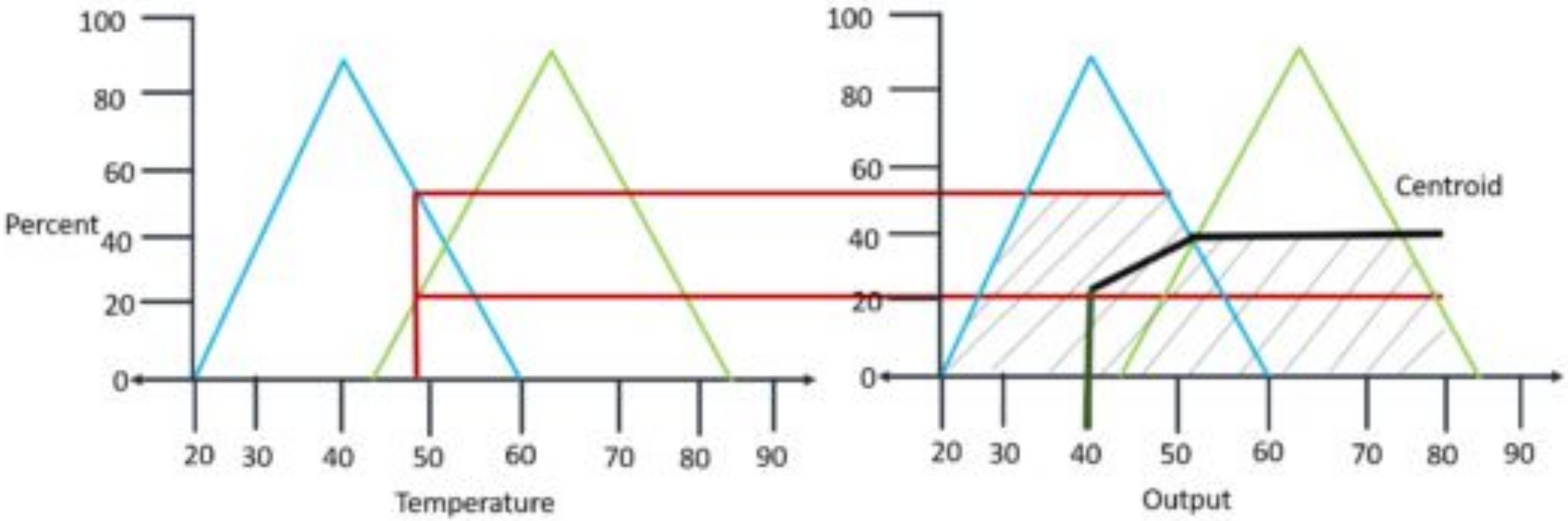

100
80
60
Percent
40
20
0
Cold
Normal
Hot
20
30
40
50
60
70
80
90
100
Temperature

- **Step 2**
- In the next step, we will use three fuzzy sets for the output, **Slow, Medium** and **Fast**. A set of functions is created for each output set just as for the input sets.



100
80
60
Percent
40
20
0
Slow
Medium
Fast
20
30
40
50
60
70
80
90
100
Speed

- **Step 3**
- Now that we have our membership functions defined, we can create the rules that will define how the membership functions will be applied to the final system. We will create three rules for this system.
  - **If Hot then Fast**
  - **If Warm then Medium**
  - **And, if Cold then Slow**



100
80
60
Percent
40
20
0
20
30
40
50
60
70
80
90
Temperature
100
80
60
40
20
0
20
30
40
50
60
70
80
90
Output
Centroid

- These rules apply to the membership functions to produce the crisp output value to drive the system. Thus, for an input value of **52 degrees**, we intersect the membership functions. Here, we are applying two rules as the intersection occurs on both functions.
- You can extend the intersection points to the output functions to produce an intersecting point. You can then truncate the output functions at the height of the intersecting points.

Problem :

Consider two fuzzy subsets of the set $X = \{a, b, c, d, e\}$ defined by

$$A = \{1/a, 0.3/b, 0.2/c 0.8/d\}, \quad B = \{0.6/a, 0.9/b, 0.1/c, 0.3/d, 0.2/e\}.$$

Compute the following: Supp (A), Supp(B), $A \cup B$, $A \cap B$, $\overline{A}$ and $\overline{B}$.