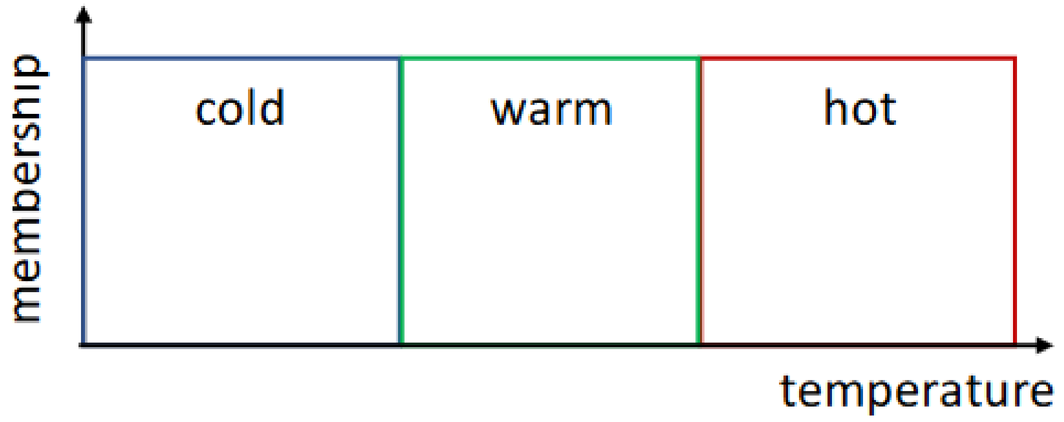
Fuzzy Rules and Reasoning

Human Reasoning and Language

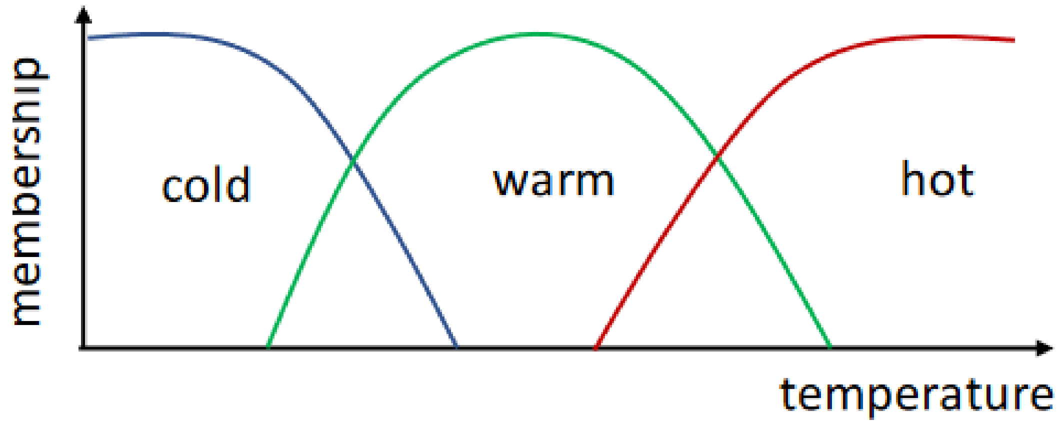
- The classifications used in language are often vague. What does a term like Large, Fast, Hot or Very Hot mean?
- Fuzzy logic is used to capture this vagueness
- Descriptions of solutions to difficult problems can be provided by experts (user) and then encoded using fuzzy logic.
- Fuzzy logic provides us with a way of capturing these Linguistic Variables

Fuzzy set

- Fuzzy sets provide the mechanism to capture linguistic variables and record vagueness
- They do this by allowing non-binary membership of a set



Crisp sets – Binary Membership



Fuzzy sets – continuous values of membership

Membership function

- One of the key issues in all fuzzy sets is how to determine fuzzy membership functions
- Membership functions can

– either be chosen by the user arbitrarily or based on the user's experience (MF chosen by two users could be different depending upon their experiences, expertise, perspectives, etc.)

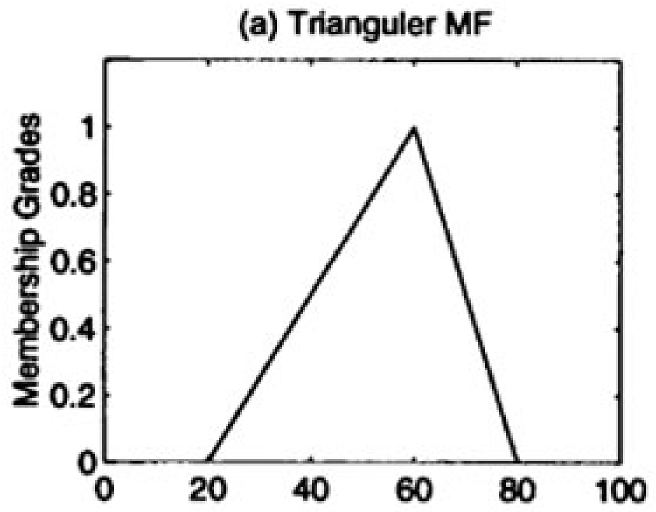
– Or be designed using machine learning methods (e.g., artificial neural networks, genetic algorithms, etc.)

Membership function

- The membership function describes the membership value of a variable for a fuzzy set
- Membership functions may be of any form, however there are standard forms such as *triangular, trapezoidal, sigmoid , bell-shaped, gaussian functions* etc.
- Triangular and trapezoidal functions are more popular as they are less computationally demanding. They may be used as approximations to more complex functions
- A crisp set has a rectangular membership function

Membership Function of one dimension:

- A ***triangular*** membership function is specified by three parameters {a, b, c}:



triangle(x; a, b, c) = 0 if x $\leq$ a;

= (x-a)/(b-a) if a $\leq$ x $\leq$ b;

= (c-b)/(c-b) if b $\leq$ x $\leq$ c;

= 0 if c $\leq$ x.

OR

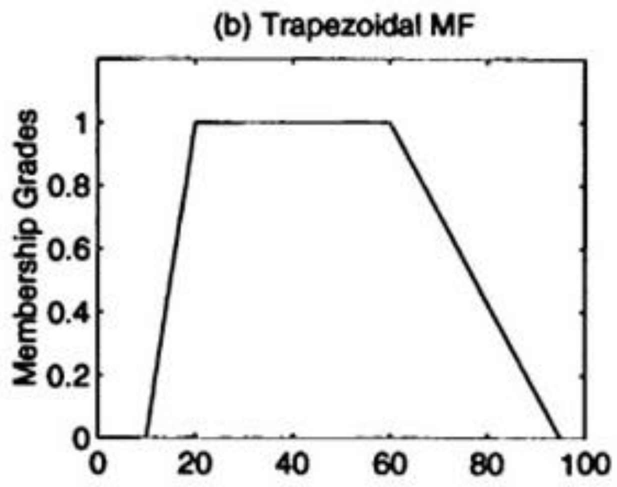
$$\boldsymbol{triangle(x; a, b, c) = max\left(min\left(\frac{(x-a)}{(b-a)}, \frac{(c-x)}{(c-b)}\right), 0\right)}$$

(with $a < b < c$)

Membership Function of one dimension:

- A ***trapezoidal*** membership function is specified by four parameters {a, b, c, d} as follows:

trapezoid(x; a, b, c, d) = 0 if x ≤ a;
= (x-a)/(b-a) if a ≤ x ≤ b;
= 1 if b ≤ x ≤ c;
= (d-x)/(d-c) 0 if c ≤ x ≤ d;
= 0, if d ≤ x.



OR

$$\boldsymbol{trapezoid(x; a, b, c, d) = max(min\left(\frac{(x-a)}{(b-a)}, 1, \frac{(d-x)}{(d-c)}\right), 0)}$$

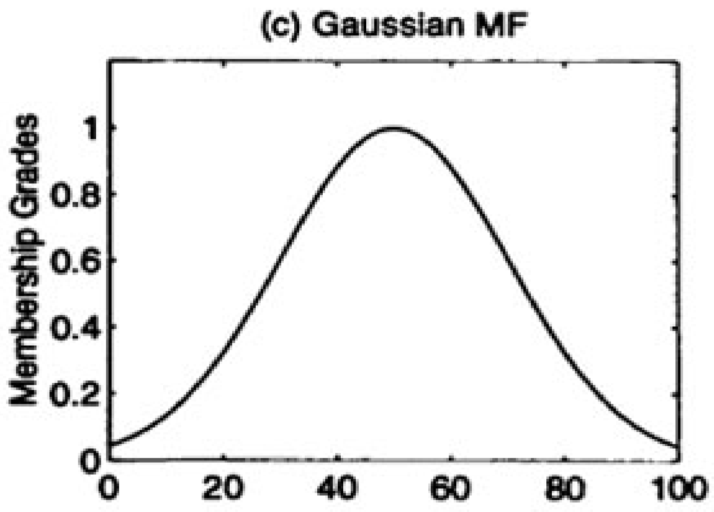
(with a < b <= c < d)

Membership Function of one dimension:

- A Gaussian MF is specified by two parameters{c, σ}:

$$gaussian(x; \mu, \sigma) = e^{-\frac{1}{2}(\frac{(x-c}{\sigma})^2}$$

- Where c represents the MFs center and σ determines the MFs width.

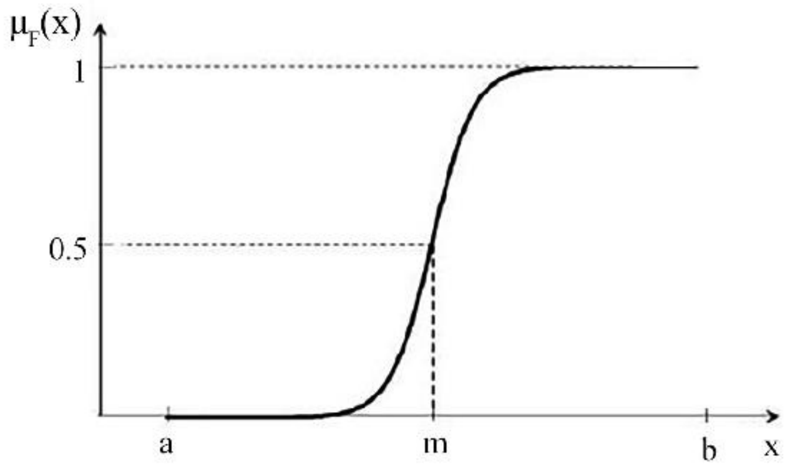


Membership Function of one dimension:

- A Sigmoid MF is specified by two parameters { a, c }:

$$sig(x; a, c) = \frac{1}{1 + e^{[-(x-c)]}}$$

- Where a controls the slope at the crossover point x=c.

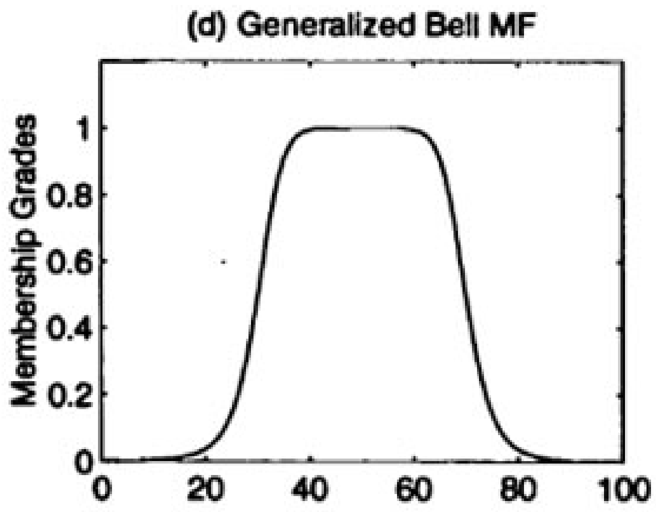


Membership Function of one dimension:

- A Generalized bell MF (or **bell MF** or **Cauchy MF**) is defined by three parameters{a,b,c}:

$$\boldsymbol{bell(x; a, b, c) = \frac{1}{1 + \left|\frac{x - c}{a}\right|^{2b}}}$$

Where the parameter b is usually positive.



Membership Function of one dimension:

- **Z-shaped membership function:** $\mu_{\underset{\sim}{A}}(x) = \begin{cases} 0, & x > d \\ \frac{d-x}{d-c}, & c \leq x \leq d \\ 1, & x < c \end{cases}$ where $c \leq d$

- **S-shaped membership function:** $\mu_{\underset{\sim}{A}}(x) = \begin{cases} 0, & x < a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ 1, & x > b \end{cases}$ where $a \leq b$

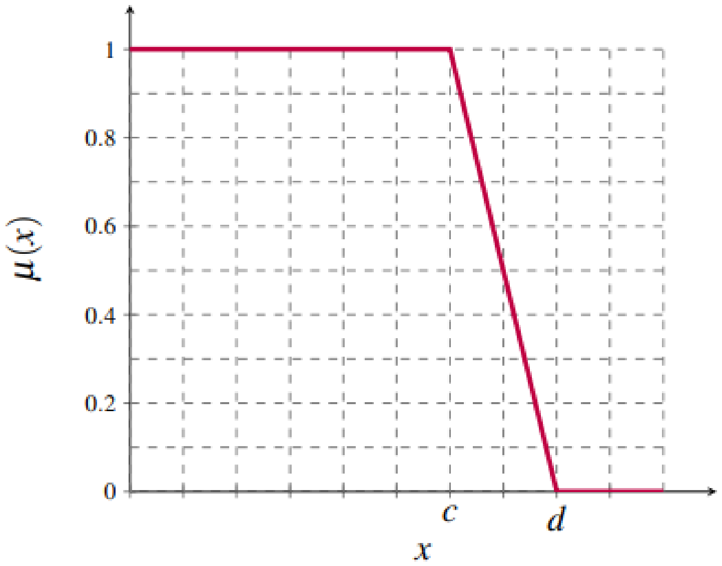


Figure 19: Z-shaped membership function.

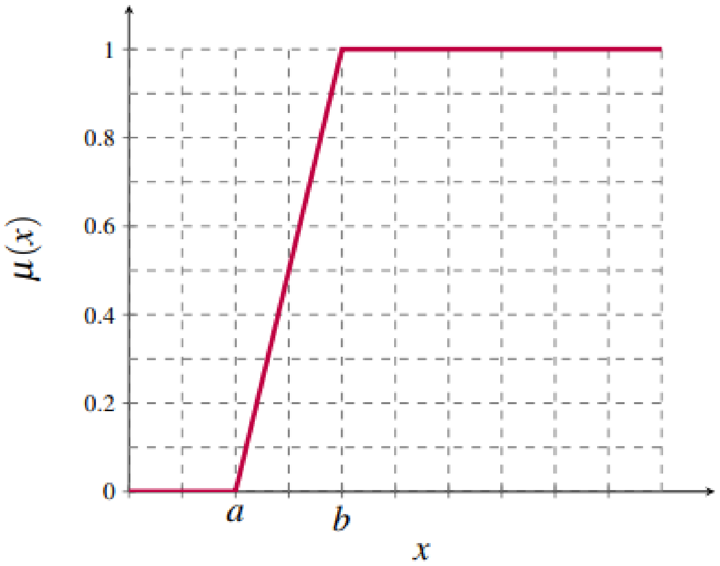


Figure 20: S-shaped membership function.

Membership Function of one dimension:

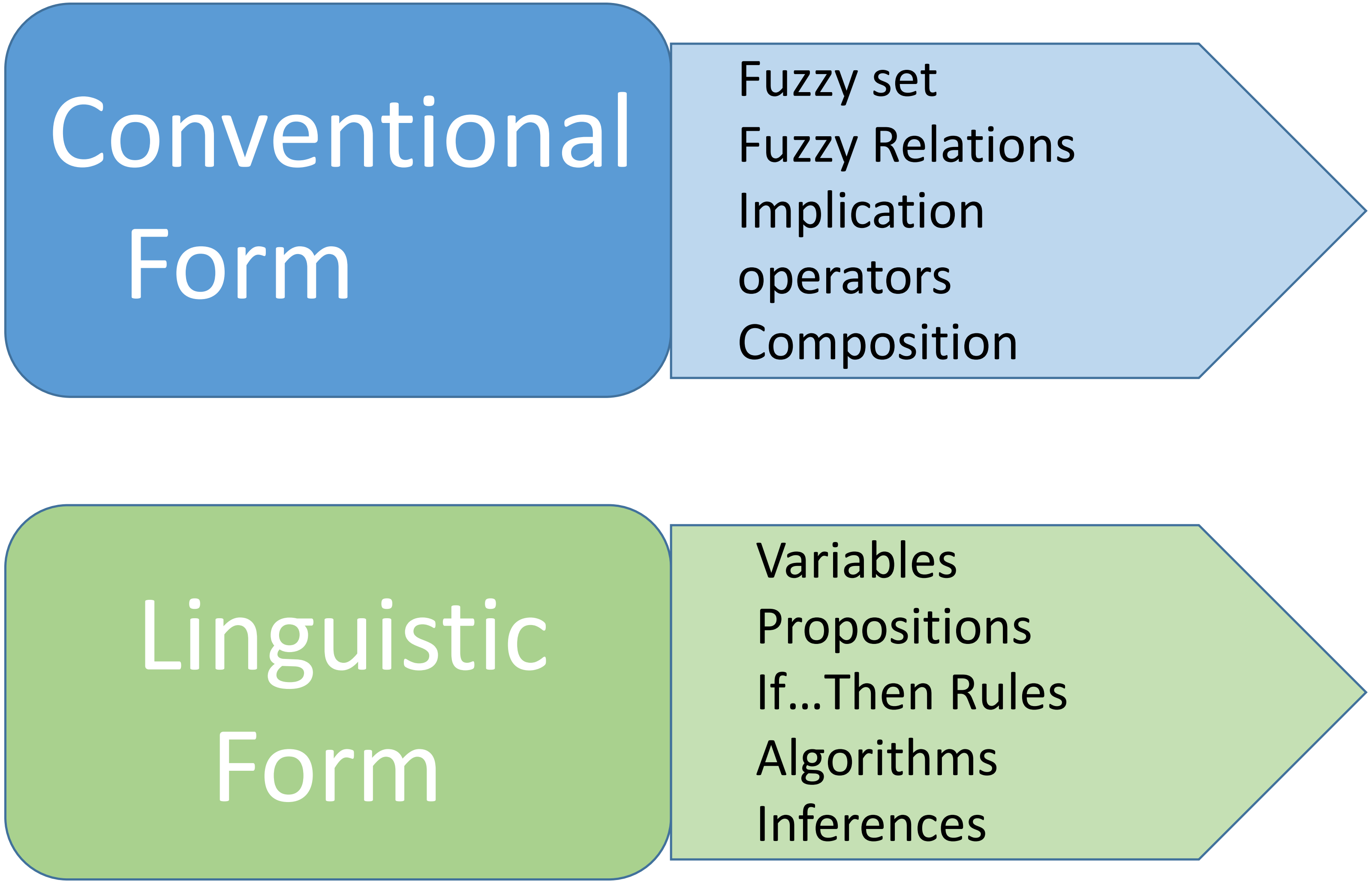
- Left-Right MF(L-R MF): The Left –Right MF or L-R MF is specified by three parameters{α, β, c}

$$LR(x; c, \alpha, \beta) = F_L\left(\frac{c - x}{\alpha}\right), x \leq c$$

$$= F_R\left(\frac{x - c}{\beta}\right), x \geq \mathrm{c}$$

Where $F_l(x)$ and $F_R(x)$ are monotonically decreasing functions defined on [0, ∞) with $F_l(0) = F_R(0) = 1$ and $\lim_{x\to\infty} F_R(x) = \lim_{x\to\infty} F_L(x) = 0$

Motivation:
Conventional techniques for system analysis are intrinsically unsuited for dealing with systems based on human judgment, perception & emotion.



Linguistic variables

- If a variable can take words in natural languages as its values, it is called a linguistic variable, where the words are characterized by fuzzy sets defined in the universe of discourse in which the variable is defined.
- Linguistic variables are used in ordinary daily activities.

- For example '*Speed*' is a linguistic variable that can take values as *slow, fast, very slow, very fast* and so on.
- This linguistic variable is made up of a number of words (linguistic terms or linguistic values) associated with degree/grade of membership values

Linguistic variables

- For example, the values of the fuzzy variable *height* could be *tall, very tall, very very tall, somewhat tall, not very tall, tall but not very tall, quite tall, more or less tall.*
- If *age* is a linguistic variable then its term set is

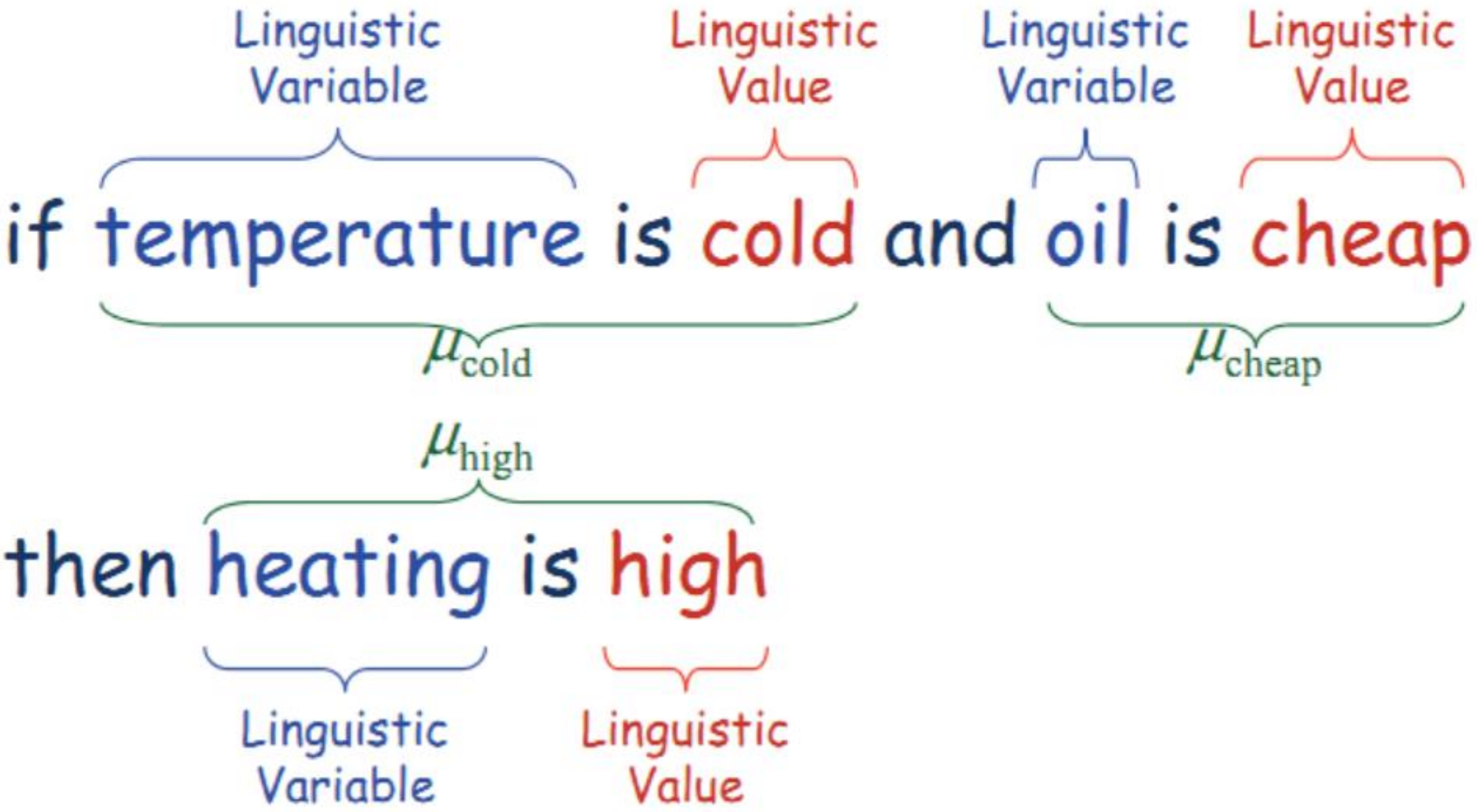
T(age) = { young, not young, very young, not very young,…… middle aged, not middle aged,… old, not old, very old, more or less old, not very old,…not very young and not very old,…}.

Example

if temperature is cold and oil is cheap

then heating is high

Example

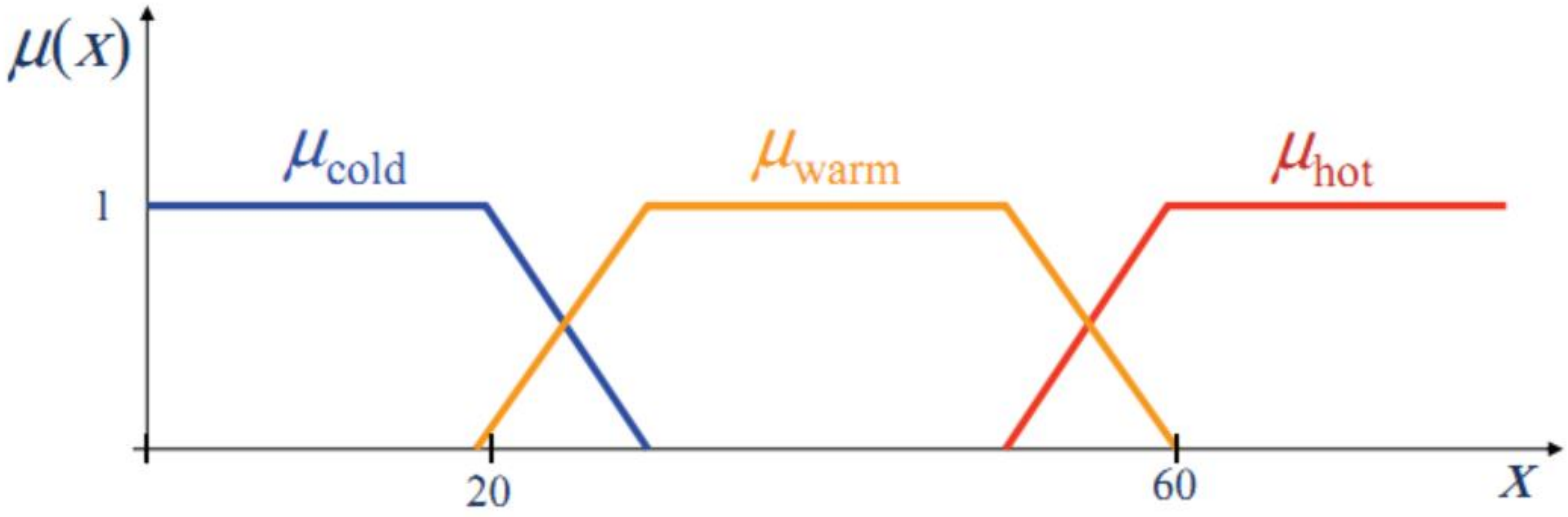


Linguistic Variable
Linguistic Value
Linguistic Variable
Linguistic Value
if temperature is cold and oil is cheap
μ_{cold}
μ_{cheap}
μ_{high}
then heating is high
Linguistic Variable
Linguistic Value

Example

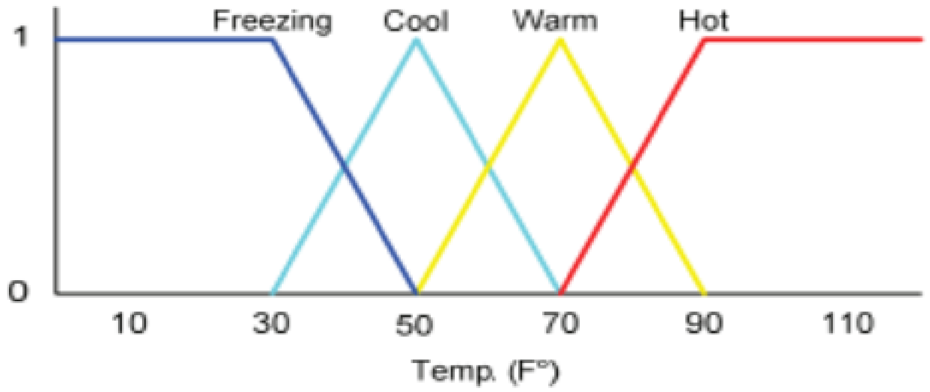
Linguistic Variable : temperature

Linguistics Terms (Fuzzy Sets) : {cold, warm, hot}

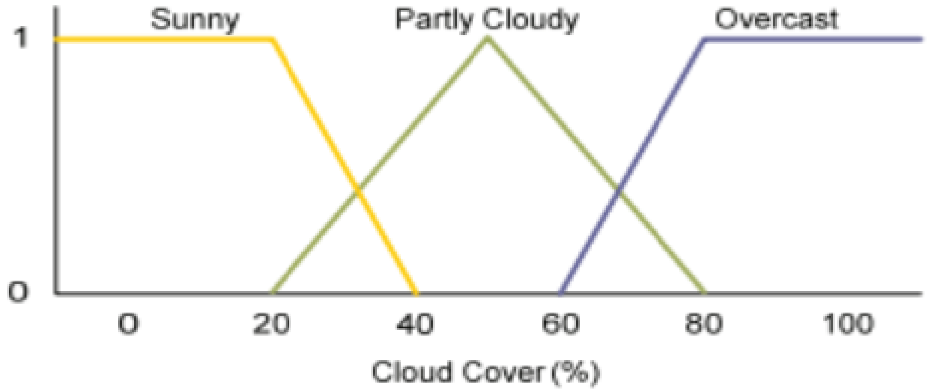


Inputs: Temperature and Cloud Cover

- Temp: {Freezing, Cool, Warm, Hot}

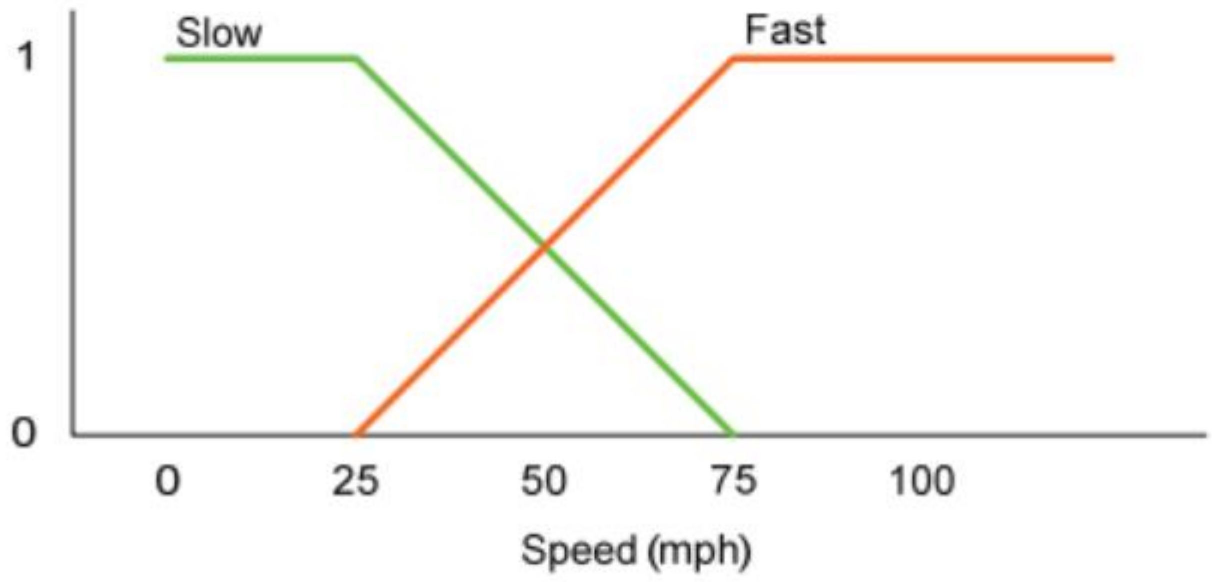


- Cover: {Sunny, Partly, Overcast}



Output: Speed

Speed: {Slow, Fast}



Rules:

- If it's Sunny and Warm, drive Fast

  $Sunny(Cover) \wedge Warm(Temp) \Rightarrow Fast(Speed)$

- If it's Cloudy and Cool, drive Slow

  $Cloudy(Cover) \wedge Cool(Temp) \Rightarrow Slow(Speed)$

- Driving Speed is the combination of output of these rules...

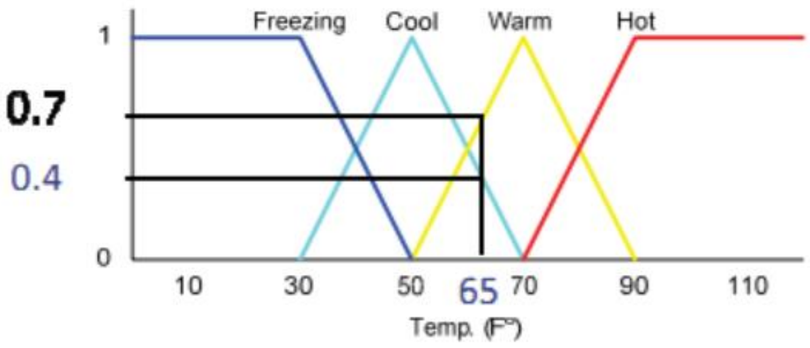
Calculate:

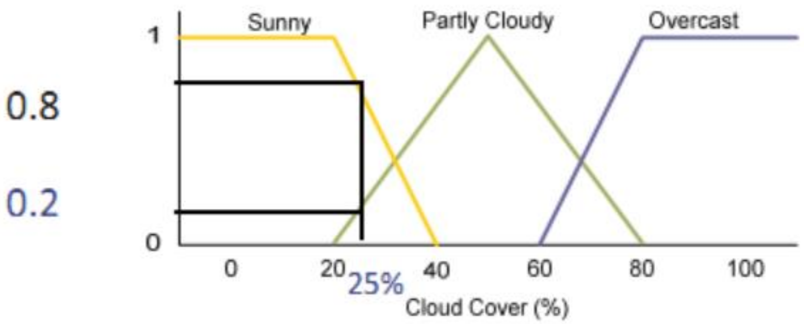
How fast will I go if it is

-65° F

- 25% cloud cover?

25% Cover ⇒Sunny = 0.8, Cloudy = 0.2





- 65 F° ⇒ Cool = 0.4, Warm= 0.7

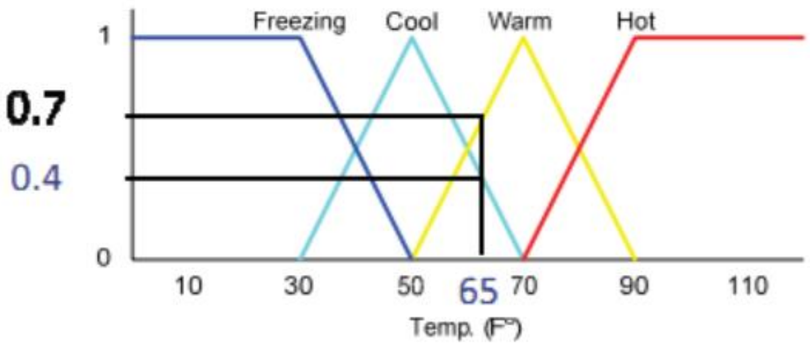
Calculate:

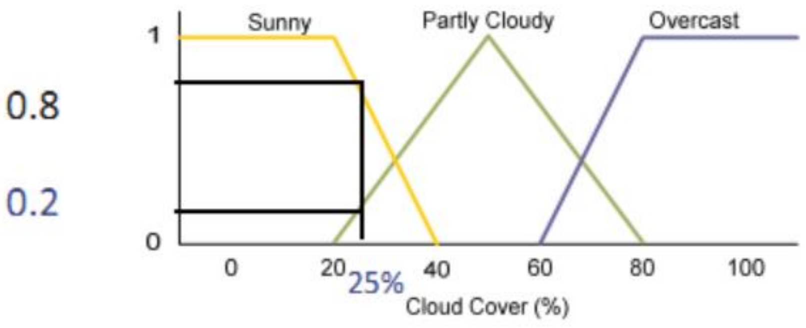
How fast will I go if it is

-65° F

- 25% cloud cover?

25% Cover ⇒Sunny = 0.8, Cloudy = 0.2





- 65 F° ⇒ Cool = 0.4, Warm= 0.7

Calculate:

- If it's Sunny and Warm, drive Fast

$$Sunny(Cover) \wedge Warm(Temp) \Rightarrow Fast(Speed)$$

$$0.8 \wedge 0.7 = 0.7$$

$$\Rightarrow \textbf{Fast} = 0.7$$

- If it's Cloudy and Cool, drive Slow

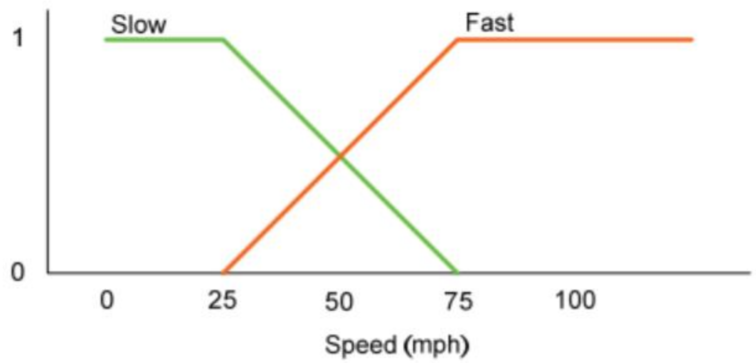
$$Cloudy(Cover) \wedge Cool(Temp) \Rightarrow Slow(Speed)$$

$$0.2 \wedge 0.4 = 0.2$$

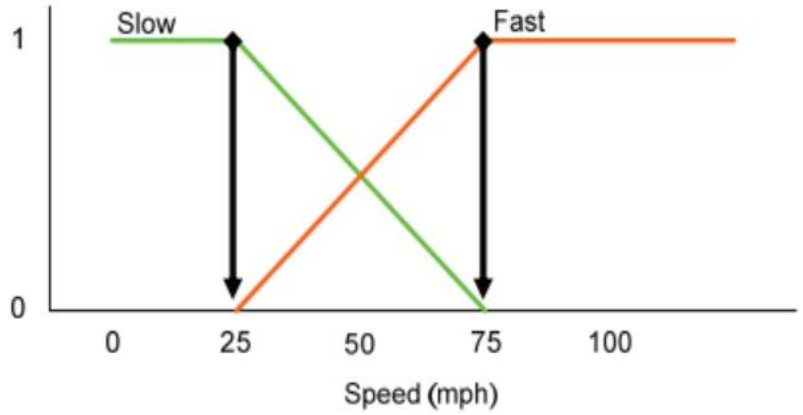
$$\Rightarrow \textbf{Slow} = 0.2$$

Constructing output

- Speed is 20% Slow and 70% Fast



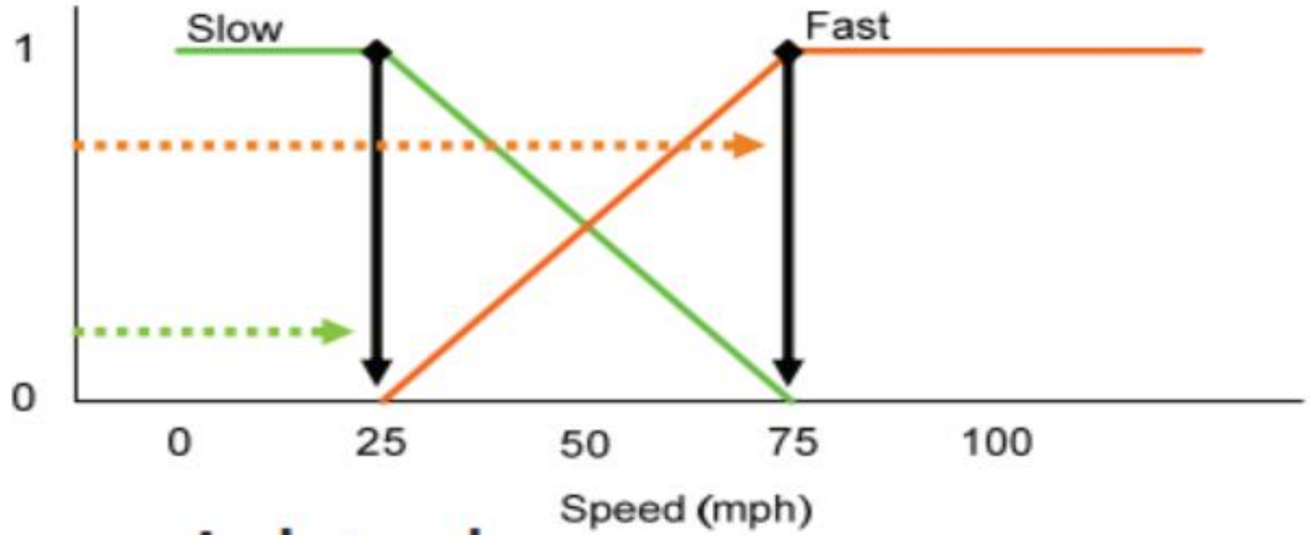
- Speed is 20% Slow and 70% Fast



- Find centroids: Location where membership is 100%

Constructing output

- Speed is 20% Slow and 70% Fast



- Speed = weighted mean

$$= (0.2*25+0.7*75)/(0.9)$$

$$= 63.8 \text{ mph}$$

Linguistic variable

- A linguistic variable is characterized by: (x, T(x), X, G, M)
  - x is the name of the linguistic variable.
  - T(x) is the called Term set is the set of linguistic values/linguistic terms that *x* can take
  - X is the universe of discourse in which the linguistic variable x can take
  - M is the semantic rule that relates each linguistic value in T with a fuzzy set
  - G is a syntactic rule that generates the linguistic values

Linguistic variable

<u>Example:</u>

A numerical variable takes numerical values

Age = 65

A linguistic variable takes linguistic values

Age is old

A linguistic value is a fuzzy set

All linguistic values form a term set

T(age) = *{young, not young, very young,... middle aged, not middle aged,... old, not old, very old, more or less old,... not very young and not very old,...}*

Where each term T(age) is characterized by a fuzzy set of a universe of discourse **X** = [0,100]

Linguistic variable

Example:

The syntactic rule refers to the way the terms in T(age) are generated

The semantic rule defines the membership function of each linguistic value of the term set

The term set consists of **primary terms** as (young, middle aged, old) modified by the **negation** ("not") and/or the **hedges** (very, more or less, quite, extremely,…) and linked by **connectives** such as (and, or, either, neither,…)

Linguistic Variable

In our daily life, we use more than one word to describe a variable.

For example, we view ***"Speed"*** *of a car* as linguistic variable, then its value might be "*not slow*", "*very fast*", "*very slow*", "*slightly fast*" , "*more or less medium*" etc.

In general, the value of a linguistic variable is a composite term $x = x_1 x_2 x_3 \ldots x_n$, that is a concatenation of atomic terms $x_1 x_2 x_3 \ldots x_n$

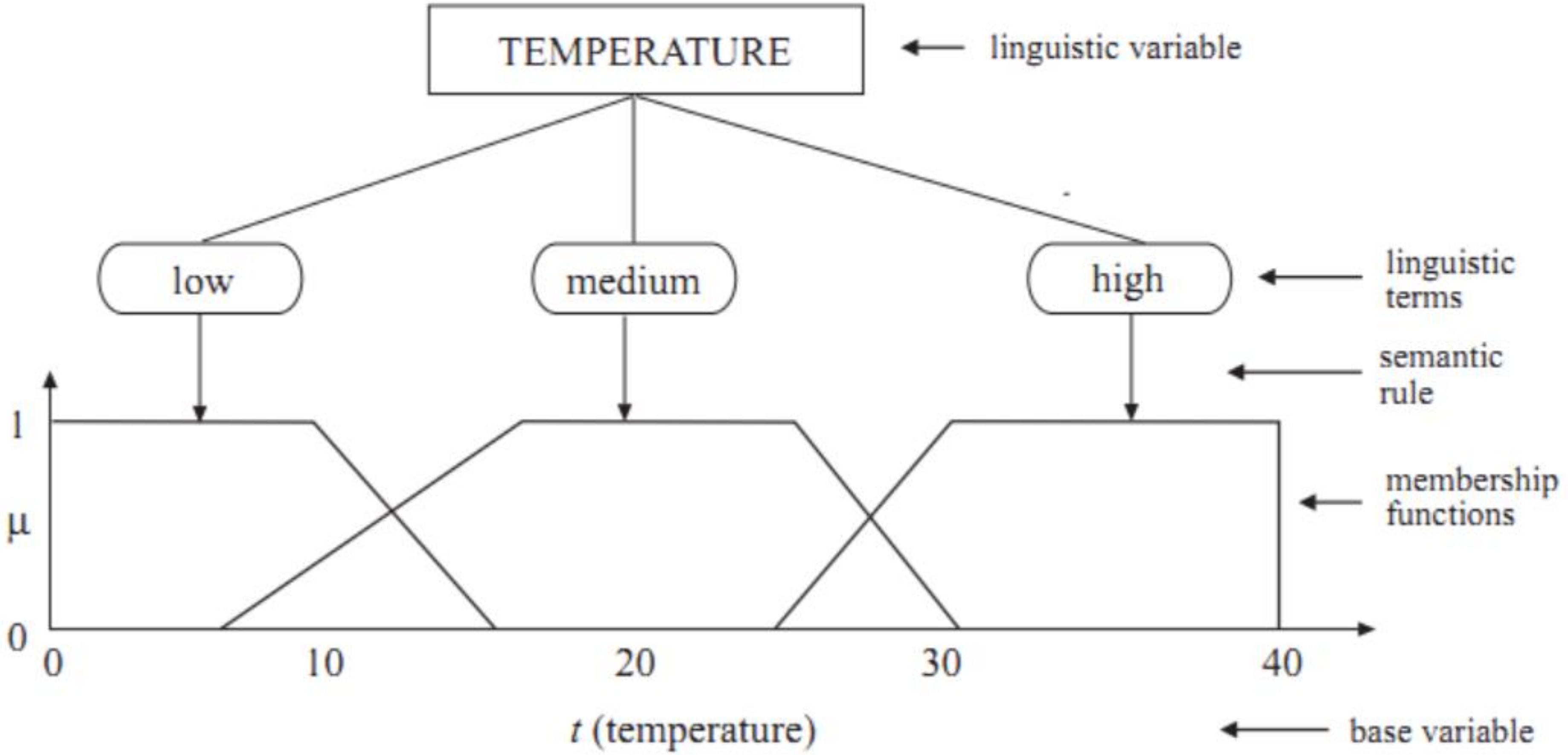
The atomic terms may be classified into three categories:

1) Primary terms: "slow" , "medium", "fast" etc.
2) Connectives ("and", "or") and complement("not")
3) Hedges : "very", "slightly", "more or less" etc.

Hedges in Linguistic variable

- A linguistic variable carries with it the concept of fuzzy set qualifiers, called *hedges*
- Hedges are terms that modify the shape of fuzz sets. They include adverbs such as *very*, *somewhat*, *quite*, *more or less* and *slightly*.

Linguistic variable



Linguistic Variable vs Membership Function

- A linguistic variable enable its value to be described both qualitatively by a linguistic term (i.e., a symbol serving as the name of a fuzzy set) and quantitatively by a corresponding membership function (which expresses the meaning of the fuzzy set).
- The linguistic term is used to express concepts and knowledge in human communication, whereas membership function is useful for processing numeric input data.

Mathematical operation modifing the meaning of a term

If A is a linguistic value then operation *concentration* is defined by CON(A) = A^2,
and *dilation* is defined by DIL(A) = $A^{0.5}$.
Using these operations we can generate linguistic hedges as shown in the following examples.

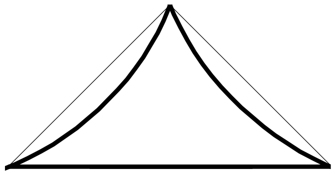
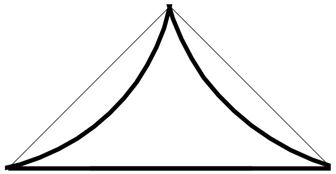
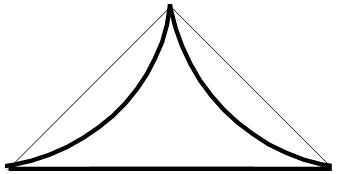
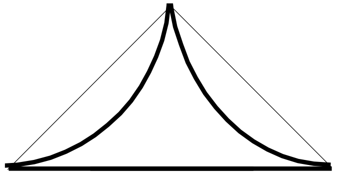
very A = con (A)
more or less A = dil(A)

Mathematical operation modifing the meaning of a term

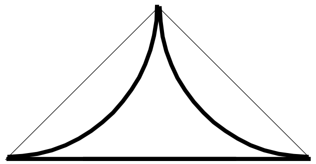
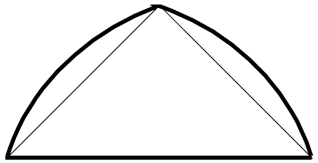
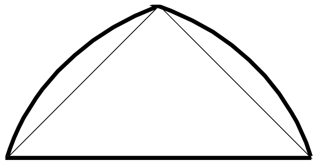
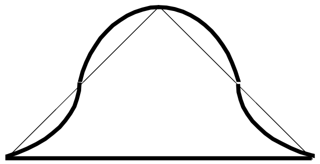
Hedges act as operations themselves, For instance.

- ***very*** performs concentration and creates a new subset. From the set of tall men, it derives the subset of very tall men. Extremely serves the same purpose to a greater extent. o
- ***more or less*** performs dilation; for example, the set of more or less tall men is broader than the set of tall men (dilation is an operation opposite to concentration. It expands the set).
- ***Indeed***, the operation of intensification, it can be done by increasing the degree of membership above 0.5 and decreasing those below 0.5.

Representation of hedges in fuzzy logic

Hedge	*Mathematical Expression*	*Graphical Representation*
A little	$[\mu_A(x)]^{1.3}$	
Slightly	$[\mu_A(x)]^{1.7}$	
Very	$[\mu_A(x)]^{2}$	
Extremely	$[\mu_A(x)]^{3}$	

Representation of hedges in fuzzy logic

Hedge	*Mathematical Expression*	*Graphical Representation*
Very very	$[\mu_A(x)]^4$	
More or less	$\sqrt{\mu_A(x)}$	
Somewhat	$\sqrt{\mu_A(x)}$	
Indeed	$2\,[\mu_A(x)]^2$ if $0 \le \mu_A \le 0.5$ $1 - 2\,[1 - \mu_A(x)]^2$ if $0.5 < \mu_A \le 1$	

Hedges examples

Suppose we define fuzzy sets *Small* and *Large* on Y={1,2,3,4,5} as:

$$Small = \frac{1}{1} + \frac{0.8}{2} + \frac{0.6}{3} + \frac{0.4}{4} + \frac{0.2}{5} \quad Large = \frac{0.2}{1} + \frac{0.4}{2} + \frac{0.6}{3} + \frac{0.8}{4} + \frac{1}{5}$$

Obtain the fuzzy sets representing:

i) *Very small*

ii) *Very very large*

Solution:

$$very\ small = \frac{1^2}{1} + \frac{0.8^2}{2} + \frac{0.6^2}{3} + \frac{0.4^2}{4} + \frac{0.2^2}{5}$$
$$= \frac{1}{1} + \frac{0.64}{2} + \frac{0.36}{3} + \frac{0.16}{4} + \frac{0.04}{5}$$

$$very\ very\ large = \frac{(.2^2)^2}{1} + \frac{(.4^2)^2}{2} + \frac{(.6^2)^2}{3} + \frac{(.8^2)^2}{4} + \frac{(1^2)^2}{5}$$
$$= \frac{.0016}{1} + \frac{.0256}{2} + \frac{.1296}{3} + \frac{.4096}{4} + \frac{1}{5}$$

Fuzzy Rules

Classical Implications

$$A \rightarrow B$$

A	B	$A \rightarrow B$
T	T	T
T	F	F
F	T	T
F	F	T

A	B	$A \rightarrow B$
1	1	1
1	0	0
0	1	1
0	0	1

$$\equiv$$

$$\neg A \cup B$$

A	B	$\neg A \cup B$
T	T	T
T	F	F
F	T	T
F	F	T

A	B	$\neg A \cup B$
1	1	1
1	0	0
0	1	1
0	0	1

Classical Implications

$$A \rightarrow B$$

$$\mu_{A\rightarrow B}(x, y) = \begin{cases} 1 & \mu_A(x) \le \mu_B(y) \\ \mu_B(y) & \text{otherwise} \end{cases}$$

A	B	$A \rightarrow B$
1	1	1
1	0	0
0	1	1
0	0	1

$$\neg A \cup B$$

$$\mu_{\neg A\cup B}(x, y) = \max\left(1 - \mu_A(x), \mu_B(x)\right)$$

A	B	$\neg A \cup B$
1	1	1
1	0	0
0	1	1
0	0	1

Modus Ponens

$$\frac{A \rightarrow B \quad \wedge \; A}{B} \qquad \frac{\neg A \cup B \quad \wedge \; A}{B} \qquad \frac{\text{If } A \text{ then } B \quad \wedge \; A \text{ is true}}{B \text{ is true}}$$

A	B	$A \rightarrow B$
1	1	1
1	0	0
0	1	1
0	0	1

Fuzzy IF-THEN Rule

$$A \rightarrow B \equiv \text{If } \underbrace{x \text{ is } A}_{\substack{\text{antecedent} \\ \text{or} \\ \text{premise}}} \text{ then } \underbrace{y \text{ is } B}_{\substack{\text{consequence} \\ \text{or} \\ \text{conclusion}}}.$$

Example

$$A \to B \equiv \text{If } x \text{ is } A \text{ then } y \text{ is } B.$$

- If *pressure* is *high*, then *volume* is *small*.
- If the *road* is *slippery*, then *driving* is *dangerous*.
- If a *tomato* is *red*, then *it* is *ripe*.
- If the *speed* is *high*, then *apply the brake* a *little*.

Fuzzy IF THEN Rules

- Fuzzy rules are useful for modeling human thinking, perception (opinion, view) and judgment.
- A fuzzy if-then rule is of the form "*If x is A then y is B*" where A and B are linguistic values defined by fuzzy sets on universes of discourse X and Y, respectively.
- "*x is A*" is called antecedent (premises) and "*y is B*" is called consequent (conclusion).

Fuzzy Rules as Relations

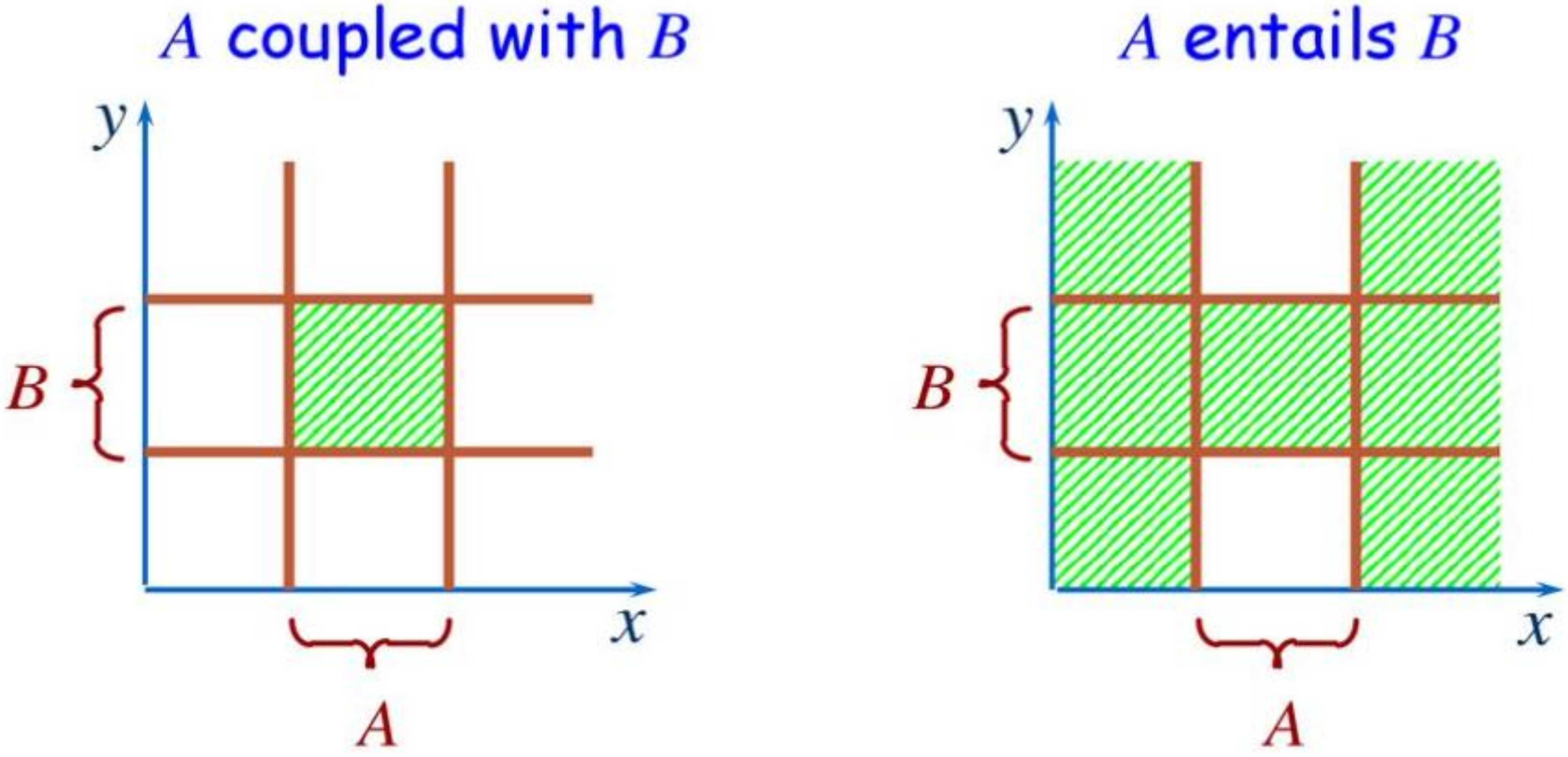
$$\underbrace{A \rightarrow B}_{R} \equiv \text{If } x \text{ is } A \text{ then } y \text{ is } B.$$

A fuzzy rule can be defined as a binary relation with MF

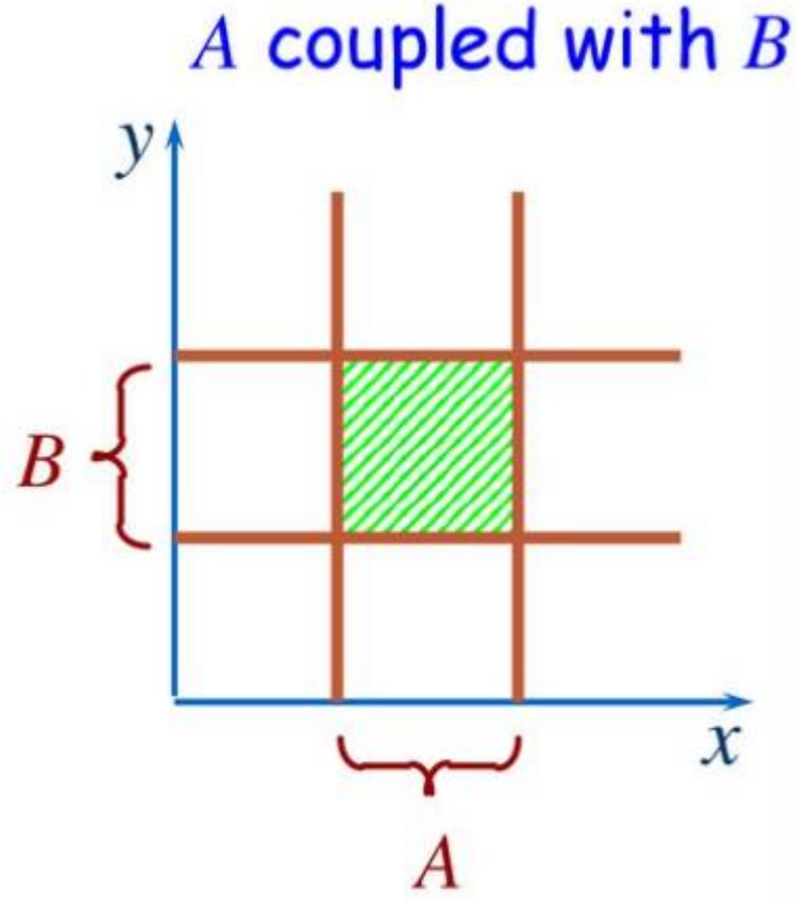
$$\underbrace{\mu_R(x, y) = \mu_{A \rightarrow B}(x, y)}_{\text{Depends on how to interpret } A \rightarrow B}$$

Depends on how to interpret $A \rightarrow B$

Interpretation of A→B



Interpretation of A→B



A coupled with B (A and B)

$$R = A \rightarrow B$$

$$= \int_{X \times Y} \mu_A(x) \tilde{*} \mu_B(y) / (x, y)$$

E.g.,

$$\mu_R(x, y) = \min(\mu_A(x), \mu_B(y))$$

Interpretation of A→B

A entails *B* (not *A* or *B*)

- Material implication

$$R = A \to B \equiv \neg A \cup B$$

- Propositional calculus

$$R = A \to B \equiv \neg A \cup (A \cap B)$$

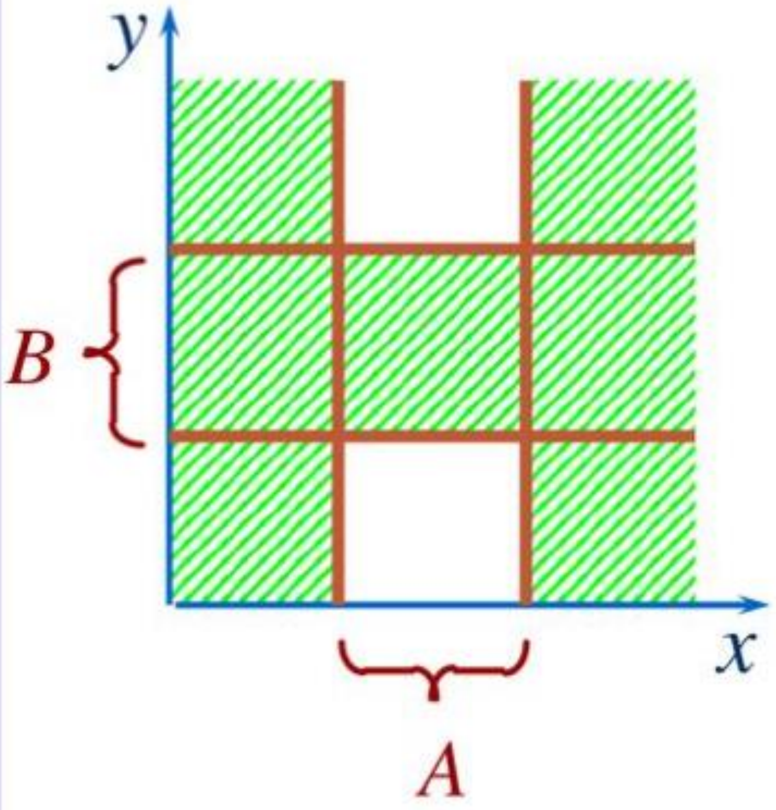
- Extended propositional calculus

$$R = A \to B \equiv (\neg A \cap \neg B) \cup B$$

- Generalization of modus ponens

$$\mu_R(x, y) = \begin{cases} 1 & \mu_A(x) \le \mu_B(y) \\ \mu_B(y) & \text{otherwise} \end{cases}$$

A entails *B*



Interpretation of A→B

A entails B (not A or B)

- Material implication

$$R = A \rightarrow B \equiv \neg A \cup B \qquad \mu_R(x,y) = \max\left(1-\mu_A(x), \mu_B(x)\right)$$

- Propositional calculus

$$R = A \rightarrow B \equiv \neg A \cup (A \cap B) \qquad \mu_R(x,y) = \max\left(1-\mu_A(x), \min\left(\mu_A(x), \mu_B(x)\right)\right)$$

- Extended propositional calculus

$$R = A \rightarrow B \equiv (\neg A \cap \neg B) \cup B \qquad \mu_R(x,y) = \max\left(1-\max\left(\mu_A(x), \mu_B(x)\right), \mu_B(x)\right)$$

- Generalization of modus ponens

$$\mu_R(x,y) = \begin{cases} 1 & \mu_A(x) \le \mu_B(y) \\ \mu_B(y) & \text{otherwise} \end{cases}$$

Fuzzy If Then rules

Example: IF profession is athlete then fitness is high

Coupling:

Athletes and only athletes have high fitness

The “If“ statement is necessary and sufficient condition

Entailment: Athletes have high fitness , Non-athelets may or may not have high fitness

The “If” statement is sufficient condition but not necessary

Fuzzy Reasoning (Approximate Reasoning)

Fuzzy reasoning (also known as approximate reasoning) is an inference procedure that derives conclusions from a set of fuzzy if-then-rules & known facts

Single rule with single antecedent

Rule: if x is A then y is B

Fact: x is A'

Conclusion: y is B'

where A’ is close to A and B’ is close to B

Fuzzy Reasoning (Approximate Reasoning)

Single Rule with multiple antecedents:

(Rule)	If x is A and y is B then z is C
(Fact)	x is A' and y is B'
(Conclusion)	*z is C'*

Fuzzy Reasoning (Approximate Reasoning)

Multiple Rules with multiple antecedents:

(Rule1) if x is A_1 and y is B_1 then z is C_1

(Rule2) if x is A_2 and y is B_2 then z is C_2

(Fact) x is A' and y is B'

(Conclusion/Consequent) *z is C'*

The Four steps of Fuzzy reasoning

Inferences in propositional logic

In propositional logic, there are two very important inference rules, Modus Ponens and Modus Tollens

		Modus Ponens	Modus Tollens
Premise 1	$\Rightarrow$	u is X	v is not Y
Premise 2	$\Rightarrow$	if u is X then v is Y	if u is X then v is Y
Consequence	$\Rightarrow$	v is Y	u is not X
Propositional logic	$\Rightarrow$	$(X \wedge (X \rightarrow Y)) \rightarrow Y$	$((\neg Y) \wedge (X \rightarrow Y)) \rightarrow (\neg X)$

Modus Ponens

μ_X	μ_Y	$X \rightarrow Y$	$X \wedge (X \rightarrow Y)$	$(X \wedge (X \rightarrow Y)) \rightarrow Y$
1	1	1	1	**1**
1	0	0	0	**1**
0	1	1	0	**1**
0	0	1	0	**1**

Modus Tollens

μ_X	μ_Y	$\neg Y$	$X \rightarrow Y$	$(\neg Y) \wedge (X \rightarrow Y)$	$\neg X$	$((\neg Y) \wedge (X \rightarrow Y)) \rightarrow (\neg X)$
1	1	0	1	0	0	**1**
1	0	1	0	0	0	**1**
0	1	0	1	0	1	**1**
0	0	1	1	1	1	**1**

Inference rules in Fuzzy logic

Fuzzy Inferencing combines - the fact obtained from the fuzzification with the rule base, and then conducts the fuzzy reasoning process.

Two important inferring procedures are:

i. Generalized Modus Ponens (GMP)

ii. Generalized Modus Tollens (GMT)

Generalized Modus Ponens (GMP)

If x is A Then y is B

x is A'

---

y is B'

Generalized Modus Tollens (GMT)

If x is A Then y is B

y is B'

---

x is A'

Here, A , B , A' and B' are fuzzy sets

Fuzzy reasoning based on max-min composition

Let **A**, **A'**, and **B** be fuzzy sets of X, X, and **Y**, respectively.

Assume that the fuzzy implication **A → B** is expressed as a fuzzy relation **R** on *X x Y* .

Then the fuzzy set **B'** induced by "**x is A'**" and the fuzzy rule "if x is A then y is B" is defined by

Generalized Modus Ponens (GMP)

If x is A Then y is B

x is A'

---

y is B'

→

$$\mu_{B'}(y) = \max_x \min[\mu_{(A')}(x), \mu_R(x, y)]$$

or, equivalentely,

$$B' = A' \circ R = A' \circ (A \rightarrow B)$$

Fuzzy reasoning based on max-min composition

Let **A**, **A'**, and **B** be fuzzy sets of X, X, and **Y**, respectively.

Assume that the fuzzy implication **A → B** is expressed as a fuzzy relation **R** on *X x Y* .

Then the fuzzy set **A'** induced by "**x is B'**" and the fuzzy rule "if x is A then y is B" is defined by

Generalized Modus Tollens (GMT)

If x is A Then y is B

y is B'

---

x is A'

$$\mu_{A'}(x) = max[min(\mu_{B'}(y), \mu_R(x, y))]$$

Or, equivalently

$$A' = B' \circ R = B' \circ (A \rightarrow B)$$

Generalized Modus Ponens (GMP)

$$P : \textbf{If } x \textbf{ is } A \textbf{ then } y \textbf{ is } B$$

Let us consider two sets of variables x and y be

$X = \{x_1, x_2, x_3\}$ and $Y = \{y_1, y_2\}$, respectively.

Also, let us consider the following.

$A = \{(x_1, 0.5), (x_2, 1), (x_3, 0.6)\}$

$B = \{(y_1, 1), (y_2, 0.4)\}$

Then, given a fact expressed by the proposition x is A',
where $A' = \{(x_1, 0.6), (x_2, 0.9), (x_3, 0.7)\}$
derive a conclusion in the form y **is** B' (using generalized modus ponens (GMP)).

Generalized Modus Ponens (GMP)

If x is A Then y is B

x is A'

---

y is B'

We are to find $B' = A' \circ R(x, y)$ where $R(x, y) = max\{A \times B, \overline{A} \times Y\}$

$$A \times B = \begin{matrix} & y_1 & y_2 \\ x_1 & 0.5 & 0.4 \\ x_2 & 1 & 0.4 \\ x_3 & 0.6 & 0.4 \end{matrix} \quad \text{and } \overline{A} \times Y = \begin{matrix} & y_1 & y_2 \\ x_1 & 0.5 & 0.5 \\ x_2 & 0 & 0 \\ x_3 & 0.4 & 0.4 \end{matrix}$$

Note: For $A \times B$, $\mu_{A \times B}(x, y) = min(\mu_A x, \mu_B(y))$

Generalized Modus Ponens (GMP)

$$R(x, y) = (A \times B) \cup (\overline{A} \times y) = \begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \end{array} \begin{array}{c} \begin{array}{cc} y_1 & y_2 \end{array} \\ \begin{bmatrix} 0.5 & 0.5 \\ 1 & 0.4 \\ 0.6 & 0.4 \end{bmatrix} \end{array}$$

Now, $A' = \{(x_1, 0.6), (x_2, 0.9), (x_3, 0.7)\}$

Therefore, $B' = A' \circ R(x, y) =$

$$\begin{bmatrix} 0.6 & 0.9 & 0.7 \end{bmatrix} \circ \begin{bmatrix} 0.5 & 0.5 \\ 1 & 0.4 \\ 0.6 & 0.4 \end{bmatrix} = \begin{bmatrix} 0.9 & 0.5 \end{bmatrix}$$

Thus we derive that y is B' where $B' = \{(y_1, 0.9), (y_2, 0.5)\}$

Generalized Modus Tollens (GMT)

Let sets of variables x and y be $X = \{x_1, x_2, x_3\}$ and $y = \{y_1, y_2\}$, respectively.

Assume that a proposition **If** x **is** A **Then** y **is** B given where $A = \{(x_1, 0.5), (x_2, 1.0), (x_3, 0.6)\}$ and $B = \{(y_1, 0.6), (y_2, 0.4)\}$

Assume now that a fact expressed by a proposition y **is** B is given where $B' = \{(y_1, 0.9), (y_2, 0.7)\}$.

We have to determine A’

Generalized Modus Tollens (GMT)

We first calculate $R(x, y) = (A \times B) \cup (\overline{A} \times y)$

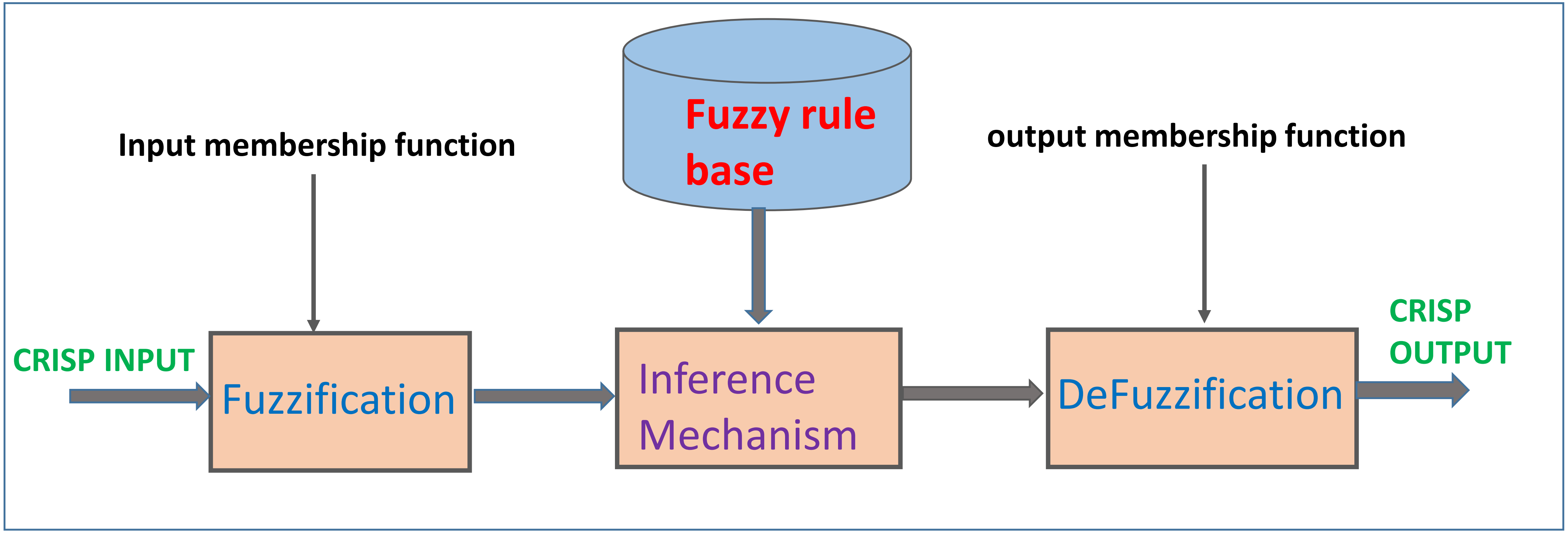
$$R(x,y) = \begin{matrix} & y_1 & y_2 \\ x_1 & 0.5 & 0.5 \\ x_2 & 1 & 0.4 \\ x_3 & 0.6 & 0.4 \end{matrix}$$

Next, we calculate $A' = B' \circ R(x, y)$

$$A' = [0.9 \quad 0.7] \circ \begin{matrix} & y_1 & y_2 \\ x_1 & 0.5 & 0.5 \\ x_2 & 1 & 0.4 \\ x_3 & 0.6 & 0.4 \end{matrix} = [0.5 \quad 0.9 \quad 0.6]$$

Hence, we calculate that x is A' where
$A' = [(x_1, 0.5), (x_2, 0.9), (x_3, 0.6)]$

Fuzzy Inference System:



Fuzzy Inference System:

Fuzzifiers: It maps the crisp (real-valued) input into a fuzzy set defined in the universe of discourse (the domain of the fuzzy set) $\mathbf{X}$ characterised by membership functions. This process is called *fuzzification*. Note: The input can also be a fuzzy set.

Knowledge Base: It is a database consisting of linguistic rules in If-Then format.

Fuzzy Inference Engine: Using the If-Then rules in *Knowledge base*, it performs reasoning by producing a fuzzy output according to the fuzzy input given by the *fuzzifier*.

Defuzzifiers: It converts the fuzzy output given by the *fuzzy inference engine* to produce a crisp (real-valued) output. This process is called *defuzzifiaction*.

Fuzzy Inference System:

Fuzzy inference engine is to produce the fuzzy output according to the crisp inputs based on the knowledge (knowledge base) represented by IF-THEN rule. This is the process of reasoning. It generally involves two processes, i.e., *rule evaluation* and *rule aggregation*

- **Rule evaluation** (implication) is to apply the fuzzy set operators (AND, OR, NOT) to the antecedents to determine the firing strength of each rule.
- **Rule aggregation** is to combine the output (consequents) fuzzy sets using the firing strengths obtained in the process of *rule evaluation*.

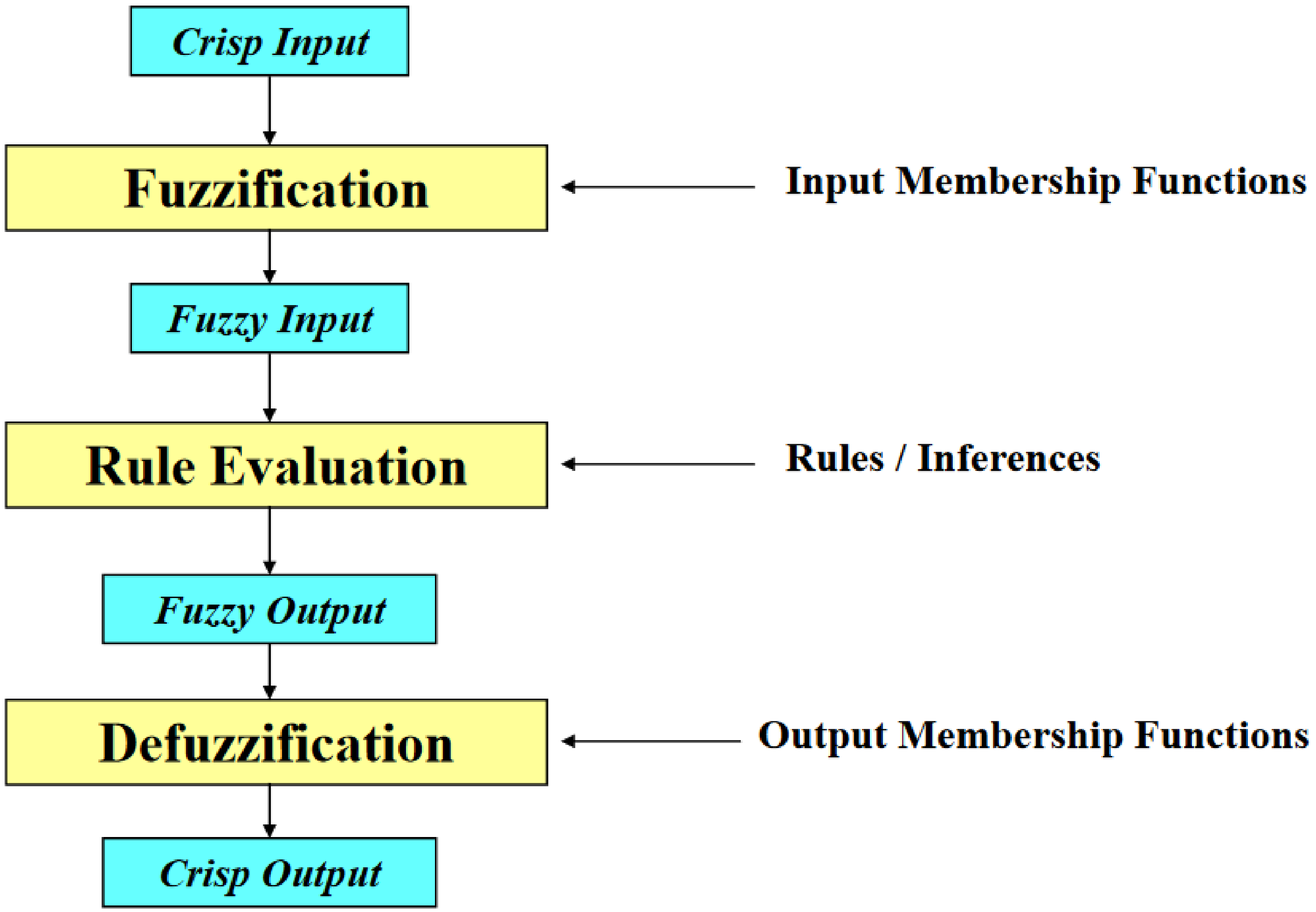
There are three standard fuzzy set operations

- *Fuzzy union* operation (*OR*), also known as *t-norm* or conjunction operator.
- *Fuzzy intersection* operation (*AND*), also known as *t-conorm*, *s-norm* operation, disjunction operation.
- *Fuzzy complement* operation (*NOT*).

Fuzzy Inference System(FIS):

Fuzzy Inference system is associated with several names:

1- Fuzzy-rule-based systems
2- Fuzzy expert systems
3- Fuzzy modelling
4- Fuzzy associative memory
5- Fuzzy logic controllers
6- Fuzzy systems.



Fuzzy Inference System

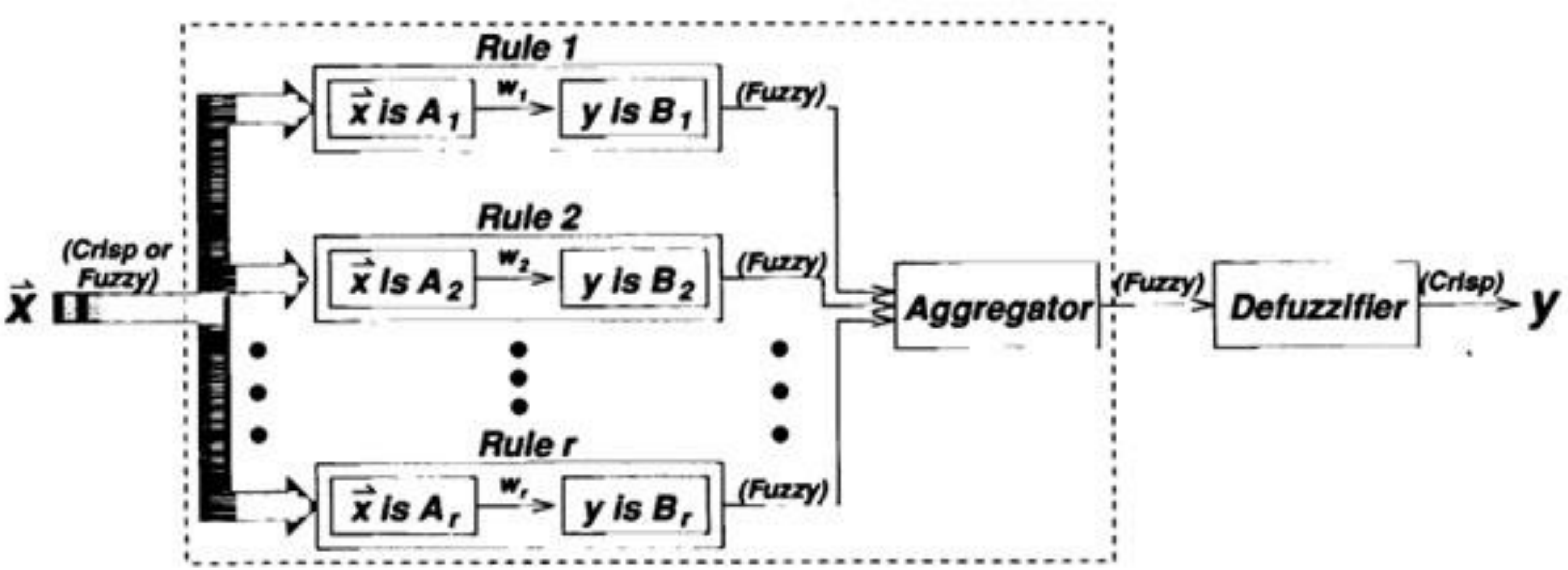


Figure: Block diagram of fuzzy inference system

Fuzzy System

Step-1: Fuzzification

- Take the crisp inputs
- Determine the degree to which these inputs belong to each of the appropriate fuzzy sets

Step-2: Rule Evaluation

If a given fuzzy rule has multiple antecedents, the fuzzy operator (AND or OR) is used to obtain a single number that represents the result of the antecedent evaluation. This number (the truth value) is then applied to the consequent membership function

Fuzzy System

Step 3: Aggregation of The Rule

- Aggregation is the process of unification of the outputs of all rules.
- We take the membership functions of all rule consequents previously clipped or scaled and combine them into a single fuzzy set

Step 4: Defuzzification

- Fuzziness helps us to evaluate the rules, but the final output of a fuzzy system has to be a crisp number.
- The input for the defuzzification process is the aggregated output fuzzy set and the output is a single number.

Inference Mechanism

(i) Mamdani fuzzy inference

(ii) Sugeno fuzzy inference or TSK (Takagi Sugeno Kang)

(ii) Tsukamoto

Inference Mechanism

- The most commonly used fuzzy inference technique is the so-called Mamdani method.
- In 1975, Professor Ebrahim Mamdani of London
- University built one of the first fuzzy systems
- To control a steam engine and boiler combination.
- He applied a set of fuzzy rules supplied by experienced human operators.

Mamdani Inference

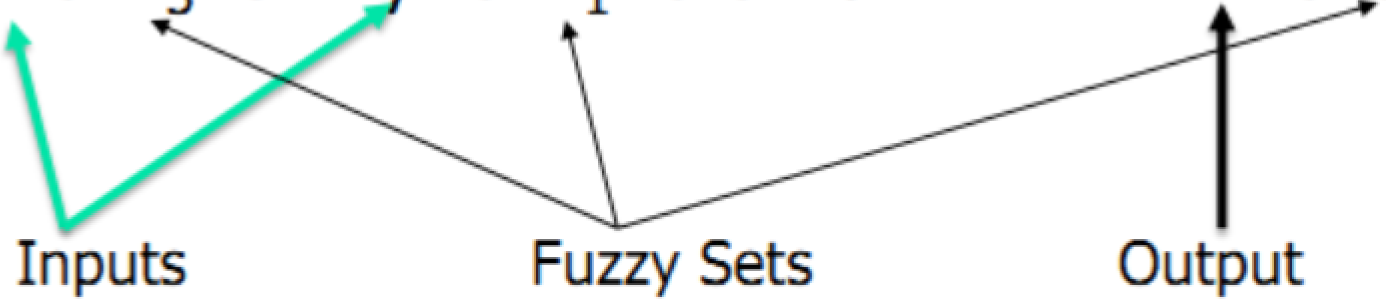
The Mamdani-style fuzzy inference process is performed in four steps:

1. Fuzzification of the input variables;
2. Rule evaluation;
3. Aggregation of the rule outputs;
4. Defuzzification.

Mamdani Inference

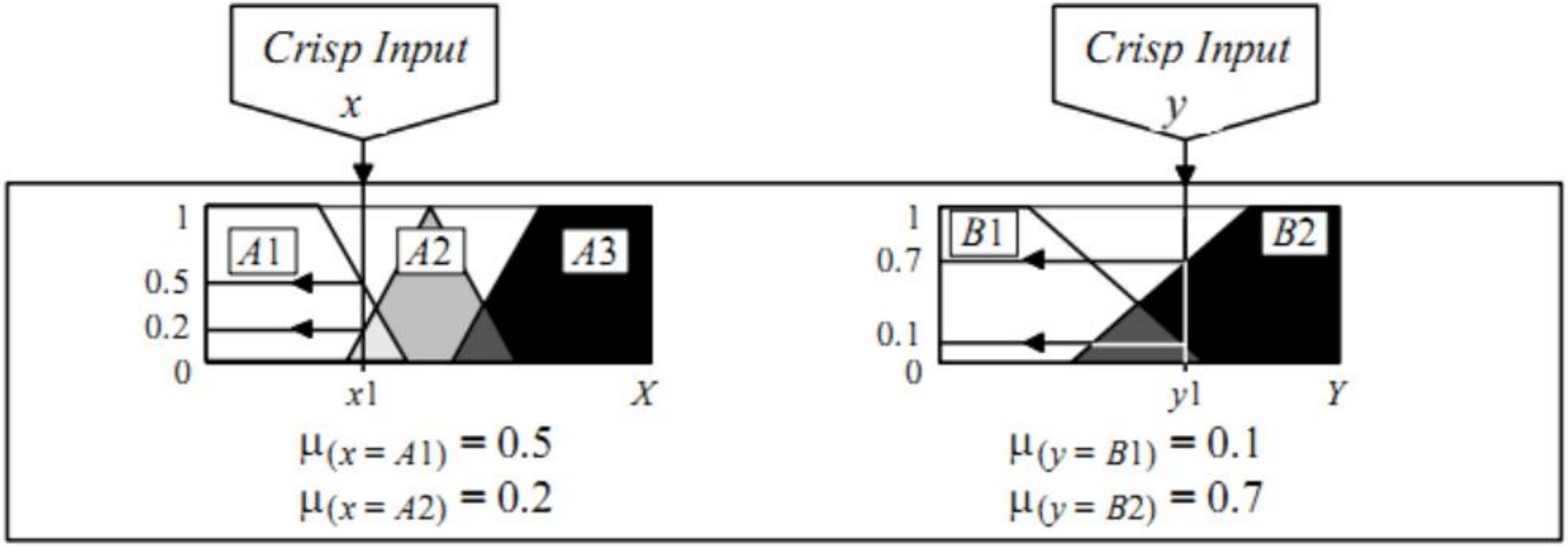
We examine a simple two-input one-output problem.

IF x is A_3 **OR** y is B_1 is small **THEN** z is C_1



Mamdani Inference

Step 1. Fuzzification: The first step is to take the crisp inputs (x and y), and determine the membership to which these inputs belong



Mamdani Inference

Step 2. Rule Evaluation: The second step is to take the fuzzified inputs, $m_{(x=A1)}=0.5$, $m_{(x=A2)}=0.2$, $m_{(y=B1)}=0.1$ and $m_{(y=B2)}=0.7$, and apply them to the antecedents of the fuzzy rules. If a given fuzzy rule has multiple antecedents, the fuzzy operator (AND or OR) is used to obtain a single number that represents the result of the antecedent evaluation. This number is then applied to the consequent membership function.

Mamdani Inference

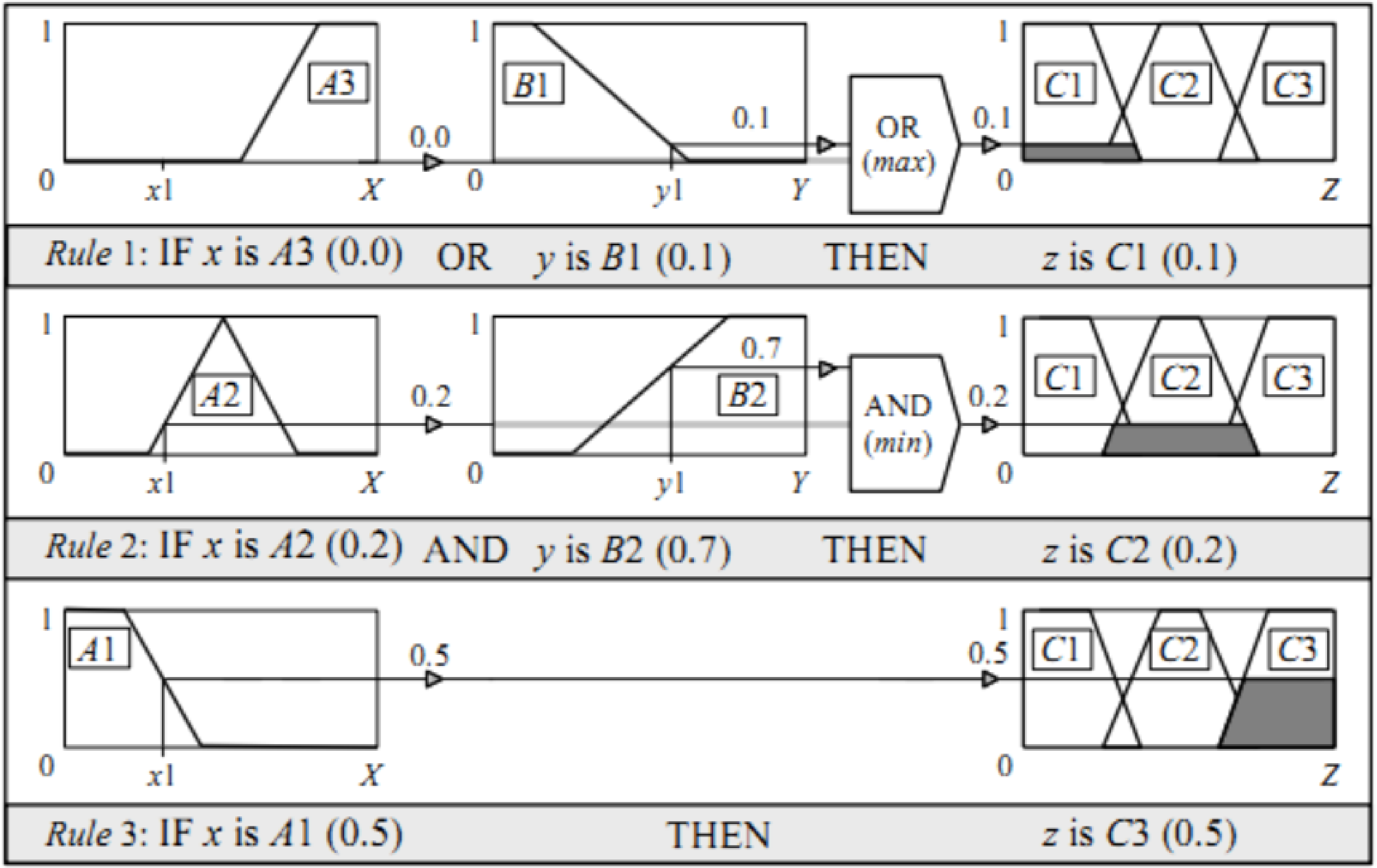
In order to evaluate the disjunction (union) of the rules, we use the OR fuzzy operation:

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

Similarly, in order to evaluate the conjunction (intersection) of the rules, we apply the AND fuzzy operation:

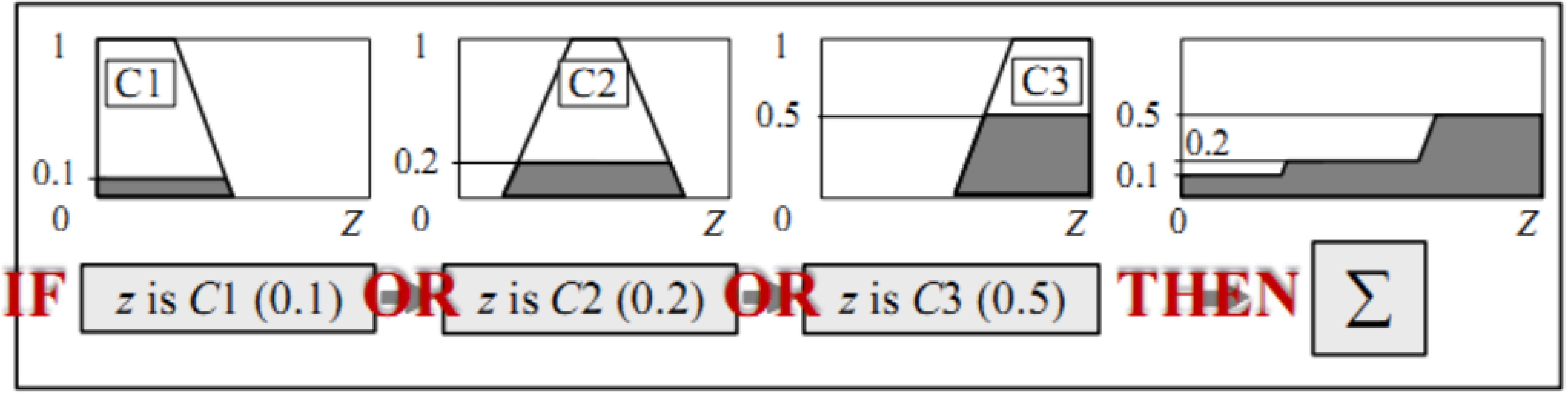
$$\mu_{A \cap B}(x) = min\ [\mu_A(x),\ \mu_B(x)]$$

Mamdani Inference



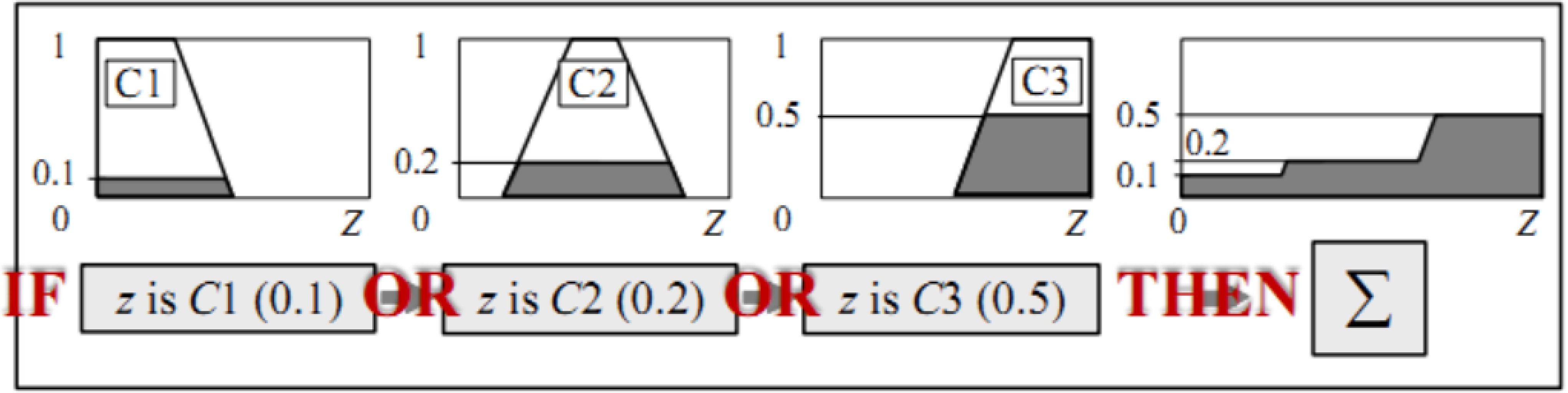
Mamdani Inference

Aggregation of the rule outputs



Mamdani Inference

Aggregation of the rule outputs



Mamdani Inference

Step 4. Defuzzification: The last step in the fuzzy inference process is defuzzification.

Fuzziness helps us to evaluate the rules, but the final output of a fuzzy system has to be a crisp number.

The input for the defuzzification process is the aggregate output fuzzy set, and the output is a single number.

Sugeno Inference

Mamdani--style inference, requires finding the centroid of a two dimensional shape by integrating across a continuously varying function. In general, this process is not computationally efficient.

Michio Sugeno suggested to use a single spike, a singleton, as the membership function of the rule .

A singleton, or more precisely a fuzzy singleton, is a fuzzy set with a membership function that is unity at a single particular point on the universe of discourse and zero everywhere else.

Fuzzy set whose support is a single point in X with:

$\mu_A(x) = 1$ is called fuzzy singleton

Sugeno Inference

It is similar to the Mamdani method in many respects. The first two parts of the fuzzy inference process, fuzzifying the inputs and applying the fuzzy operator, are exactly the same.
The main difference between Mamdani and Sugeno is that the Sugeno output membership functions are either linear or constant.

A typical rule in a Sugeno fuzzy model has the form
If *Input 1* = *x* and *Input 2* = *y*, then Output is z = ax + by + c
For a zero-order Sugeno model, the output level z is a constant (a=b =0).

Sugeno Inference

- Sugeno-style fuzzy inference is very similar to the Mamdani method.
- *Sugeno changed only a rule consequent (resultant).*

Instead of a fuzzy set, he used a mathematical function of the input variable. The format of the sugeno-style fuzzy rule is

IF x is A
AND y is B
THEN z is f (x, y)

where x, y and z are linguistic variables; A and B are fuzzy sets on universe of discourses X and Y, respectively; and f (x, y) is a mathematical function.

Sugeno Inference

- The most commonly used zero-order Sugeno fuzzy model applies fuzzy rules in the following form:

  IF x is A
  AND y is B
  THEN z is k

  where k is a constant.

- In this case, the output of each fuzzy rule is constant.
- All resultant membership functions are represented by singleton spikes.

Sugeno Inference

If x is A and y is B then $z = f(x, y)$

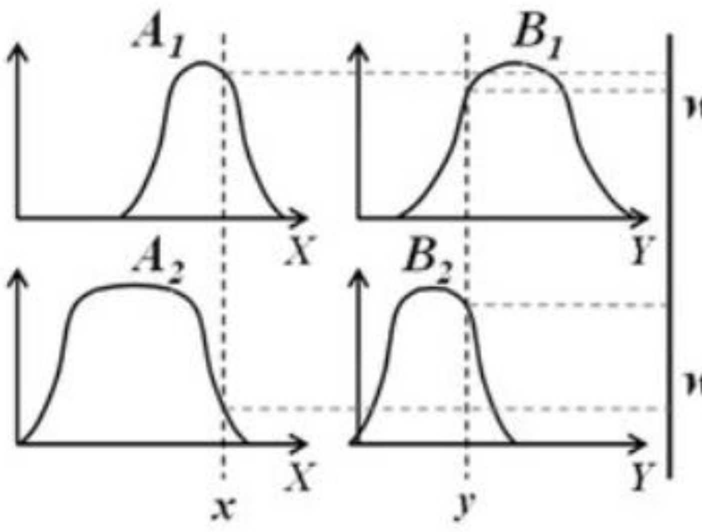
Fuzzy Sets (A, B)

Crisp Function ($z = f(x, y)$)

$f(x, y)$ is very often a polynomial function w.r.t. x and y.

Sugeno Inference

Fuzzification | Fuzzy rule | Defuzzification



w_1 $\quad f_1 = a_1 x + b_1 y + c_1$

w_2 $\quad f_2 = a_2 x + b_2 y + c_2$

$$f = \frac{w_1 f_1 + w_2 f_2}{w_1 + w_2} = \bar{w}_1 f_1 + \bar{w}_2 f_2$$

A = fuzzy set for input variable X
B = fuzzy set for input variable Y
x, *y* = input values
f = fuzzy rule
a, *b*, *c* = constant values determined by the least square method
w = membership degree of fuzzy rule
w = normalized membership degree

Sugeno Inference

Example: An example of 2-input single-output Sugeno fuzzy model with 4 rules:

Rule 1: **IF** x is $Small$ and y is $Small$ **THEN** z is $-x+y+1$

Rule 2: **IF** x is $Small$ and y is $Large$ **THEN** z is $-y+3$

Rule 3: **IF** x is $Large$ and y is $Small$ **THEN** z is $-x+3$

Rule 4: **IF** x is $Large$ and y is $Large$ **THEN** z is $x+y+2$

$$z = \frac{w_1(-x+y+1)+w_2(-y+3)+w_3(-x+3)+w_4(x+y+2)}{w_1+w_2+w_3+w_4}$$

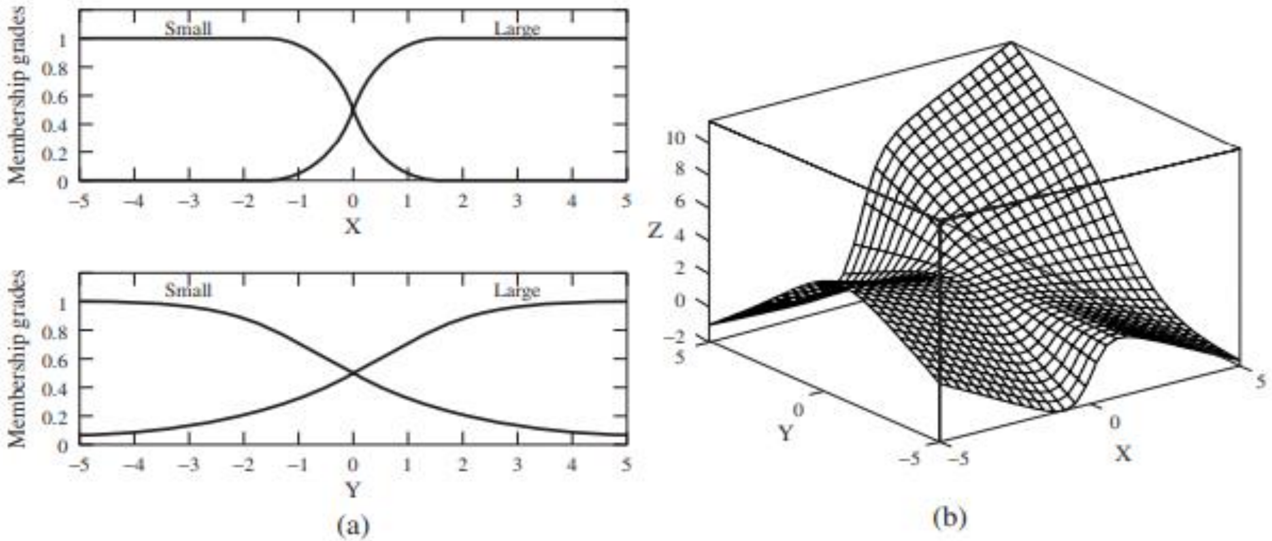
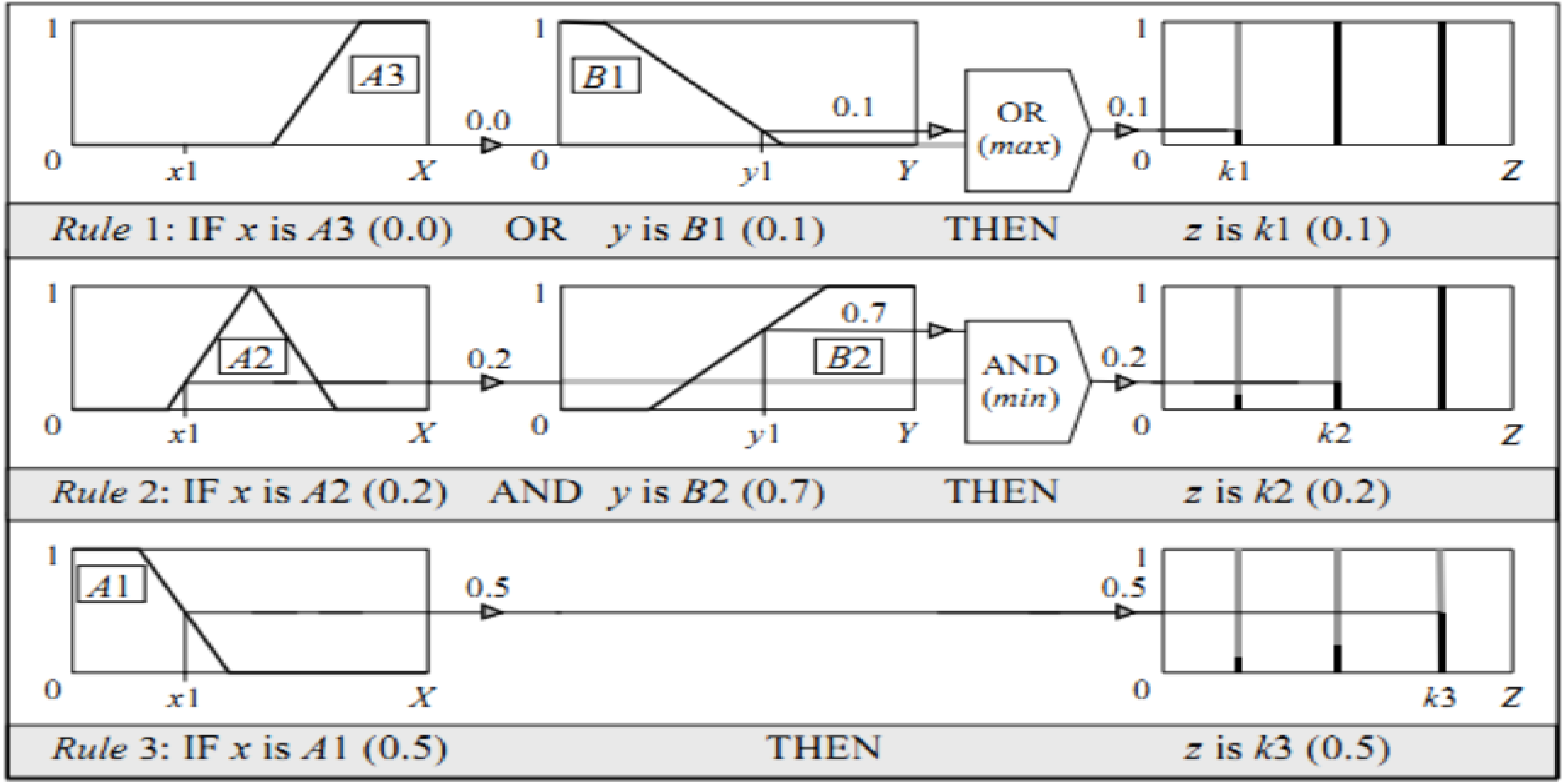


Figure 31: 2-input, single-output Sugeno fuzzy model with 4 rules. (a) Antecedent and consequent membership functions. (b) Overall output surface.

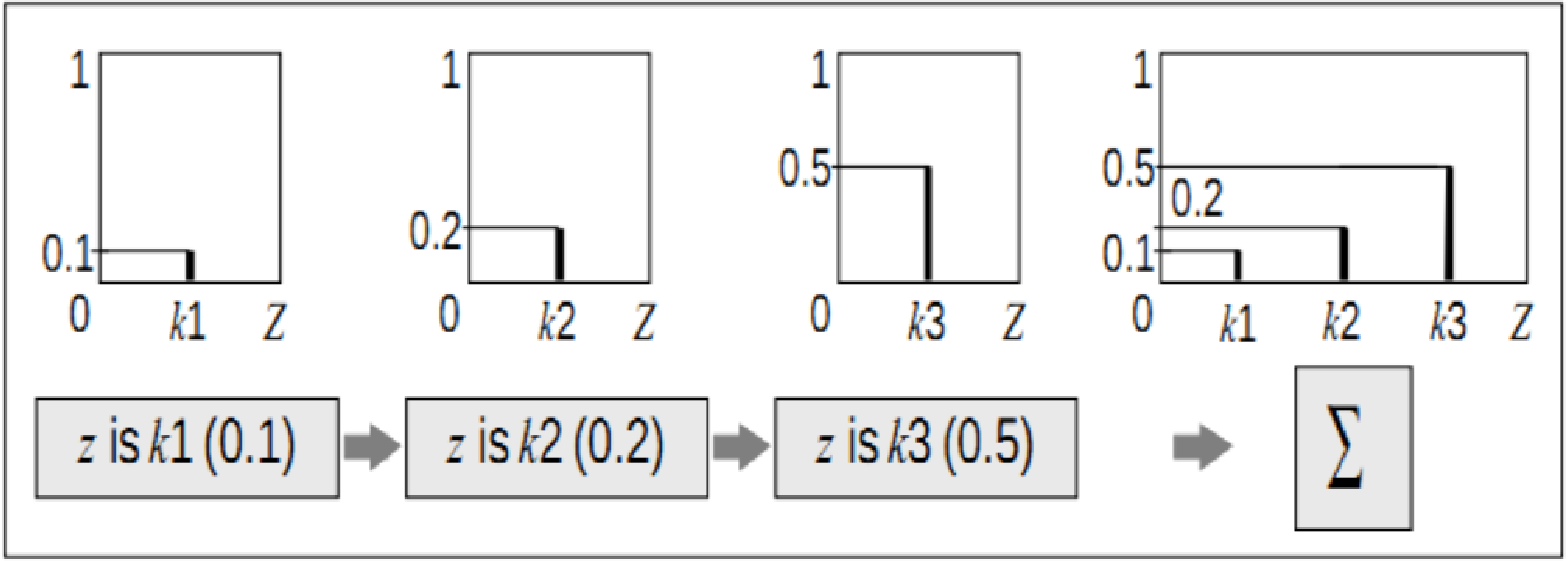
min: $w_1(x,y) = \min(\mu_{x_{Small}}(x), \mu_{y_{Small}}(y))$; $w_2(x,y) = \min(\mu_{x_{Small}}(x), \mu_{y_{Large}}(y))$; $w_3(x,y) = \min(\mu_{x_{Large}}(x), \mu_{y_{Small}}(y))$;
$w_4(x,y) = \min(\mu_{x_{Large}}(x), \mu_{y_{Large}}(y))$

product: $w_1(x,y) = \mu_{x_{Small}}(x) \times \mu_{y_{Small}}(y)$; $w_2(x,y) = \mu_{x_{Small}}(x) \times \mu_{y_{Large}}(y)$; $w_3(x,y) = \mu_{x_{Large}}(x) \times \mu_{y_{Small}}(y)$;
$w_4(x,y) = \mu_{x_{Large}}(x) \times \mu_{y_{Large}}(y)$

Sugeno Style rule evaluation



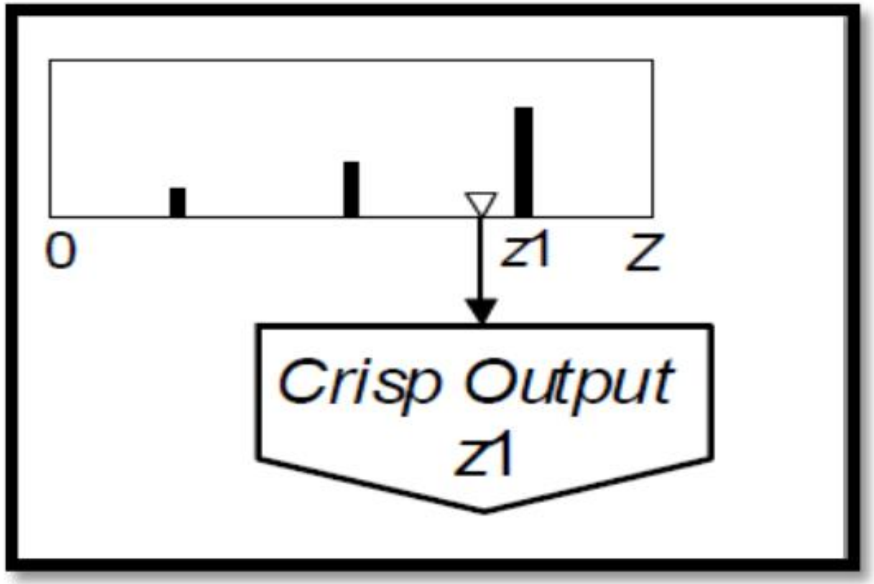
Sugeno-style aggregation of the rule outputs



Weighted average (WA):

$$WA = \frac{\mu(k1)\times k1 + \mu(k2)\times k2 + \mu(k3)\times k3}{\mu(k1)+\mu(k2)+\mu(k3)} = \frac{0.1\times 20 + 0.2\times 50 + 0.5\times 80}{0.1+0.2+0.5} = 65$$

Sugeno-style defuzzification



Mamdani vs Sugeno models

	Mamdani	Sugeno
Similarity	The antecedent parts of the rules are the same	
Difference	The consequent part fuzzy sets	The consequent part are singletons or mathematical function
Advantages	1 Easily understandable by human expert 2 Simpler to formulate rules 3 Proposed earlier and commonly used	4 More convenient in mathematical analysis 5 Guarantee continuity of the output surface
Applications	Good for capturing expertise of human operator	Good for embedding linear controller and effective when the plant model is known

Tsukamoto model

The consequent of each fuzzy if-then rule:

- A fuzzy set with a monotonical MF.
- Overall output: the weighted average of each rule's output.
- No defuzzification.
- Not as transparent as Mamdani's or Sugeno's fuzzy model.
- Not follow strictly the compositional rule of inference: the
- Output is always crisp

An example of a single-input Tsukamoto fuzzy model can be expressed as:

If X is small then Y is C1 .
If X is medium then Y is C2 .
If X is large then Y is C3

How to make a decision on which method to apply – Mamdani or Sugeno?

- Mamdani method is widely accepted for capturing expert knowledge. It allows us to describe the expertise in more intuitive, more human-like manner. However, Mamdani-type fuzzy inference entails a substantial computational burden.
- On the other hand, Sugeno method is computationally effective and works well with optimization and adaptive techniques, which makes it very attractive in control problems, particularly for dynamic nonlinear systems.

Process of developing a fuzzy expert system

1. Specify the problem and define linguistic variables.
2. Determine fuzzy sets.
3. Elicit and construct fuzzy rules.
4. Encode the fuzzy sets, fuzzy rules and to perform fuzzy inference into the expert system.
5. Evaluate and tune the system.

Tuning fuzzy system

- Review model input and output variables, and if required redefine their ranges.
- Review the fuzzy sets, and if required define additional sets on the universe of discourse. The use of wide fuzzy sets may cause the fuzzy system to perform roughly.
- Provide sufficient overlap between neighboring sets. It is suggested that triangle--to--triangle and trapezoid--to--triangle fuzzy sets should overlap between 25% to 50% of their bases.
- Review the existing rules, and if required add new rules to the base.

Tuning fuzzy system

- Examine the rule base for opportunities to write hedge rules to capture the pathological behavior of the system.
- Adjust the rule execution weights. Most fuzzy logic tools allow control of the importance of rules by changing a weight multiplier.
- Revise shapes of the fuzzy sets. In most cases, fuzzy systems are highly tolerant of a shape approximation.

Defuzzification

- Defuzzification is the process of taking the fuzzy output and deriving a single value.
- Defuzzification is the conversion of a fuzzy quantity to a precise quantity, just as fuzzification is the conversion of a precise quantity to a fuzzy quantity.

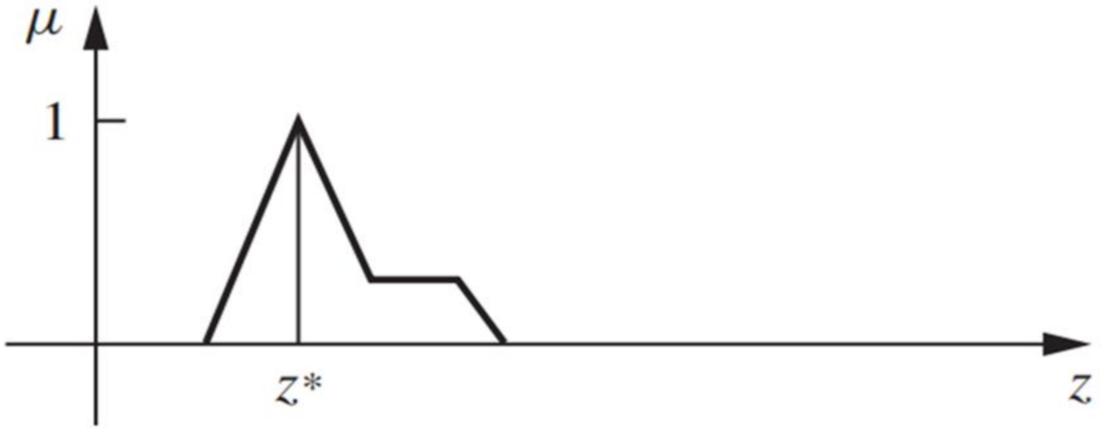
Defuzzification Methods

1. Max membership principle.
2. Centroid method.
3. Weighted average method.
4. Mean max membership.
5. Center of sums.
6. Centre of largest area.
7. First of maxima, last of maxima.
8. Lambda-cut method

Defuzzification Methods

1. Max Membership Principle

- Also known as the *height method*.
- It is limited to peaked output functions.
- $\mu_C(z^*) \geq \mu_C(z) \forall z \in Z$ where z^* is the defuzzified value.

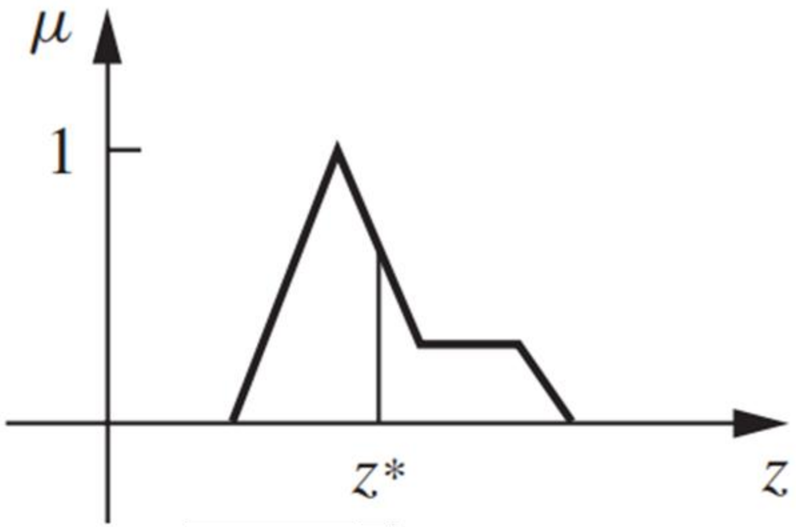


Max membership defuzzification method.

Defuzzification Methods

2. Centroid Method

- Also known as the *center of area* (COA) or *center of gravity* (COG).
- *Continuous form:* $z^* = \frac{\int \mu_C(z) z dz}{\int \mu_C(z) dz}$ where $\int$ denotes an algebraic integration.
- *Discrete form:* $z^* = \frac{\sum_{z_i \in Z} \mu_C(z_i) z_i}{\sum_{z_i \in Z} \mu_C(z_i)}$ where $\sum$ denotes an algebraic sum.



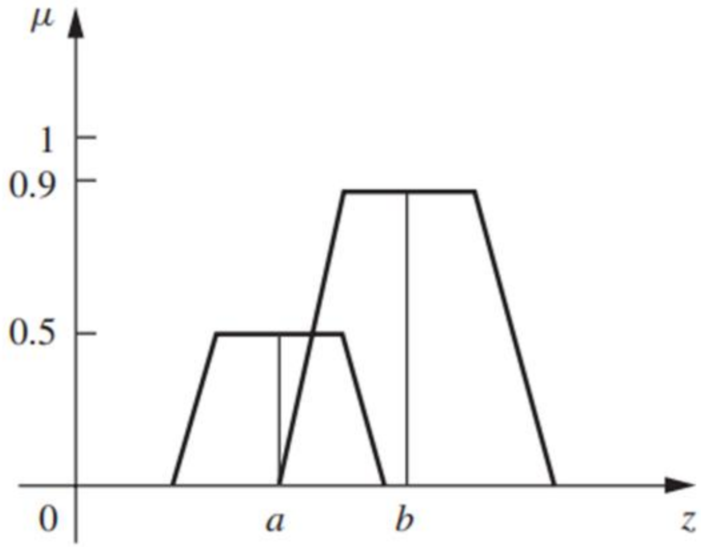
Centroid defuzzification method.

Defuzzification Methods

3. Weighted Average Method

- It is computational efficient, however, symmetrical output membership functions are required.
- $z^* = \frac{\sum \mu_C(\bar{z})\bar{z}}{\sum \mu_C(\bar{z})}$ where $\sum$ denotes an algebraic sum and $\bar{z}$ is the centroid of each symmetric inferred membership function.

Example: $z^* = \frac{0.5 \times a + 0.9 \times b}{0.5 + 0.9}$

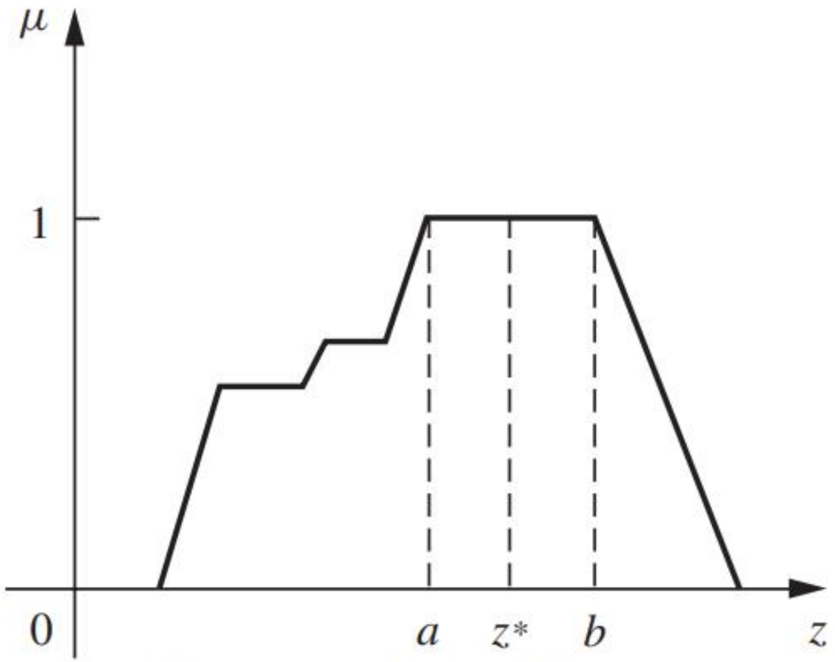


Weighted average defuzzification method.

Defuzzification Methods

4. Mean Max Membership

- Also known as *middle-of-maxima*.
- It is computational efficient.
- $z^* = \frac{a+b}{2}$.



Mean max membership defuzzification method.

Defuzzification Methods

5. Center of Sums

- Faster than many methods. Not restricted to symmetric membership functions.

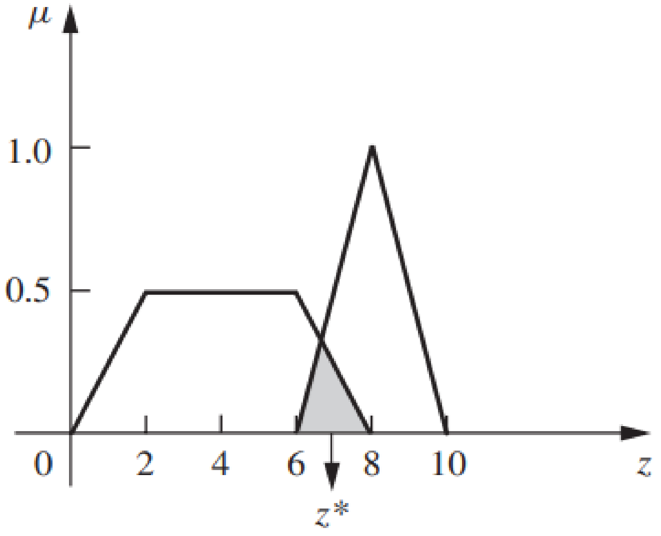
- This method finds the centroid of the individual output membership functions. The intersecting areas are included twice (*drawback*).

- *Continuous form:* $z^* = \dfrac{\sum_{k=1}^{n} \int \mu_{C_k}(z)\bar{z}_k dz}{\sum_{k=1}^{n} \int \mu_{C_k}(z) dz}$ where $\int$ denotes an algebraic integration, $\bar{z}_k$ is the centroid distance of the k^{th} inferred output membership functions.

- *Discrete form:* $z^* = \dfrac{\sum_{k=1}^{n} \sum_{z_i \in Z} \mu_{C_k}(z_i)\bar{z}_k}{\sum_{k=1}^{n} \sum_{z_i \in Z} \mu_{C_k}(z_i)}$

  where $\sum$ denotes an algebraic sum.

Example: $z^* = \dfrac{4 \times \frac{(4+8)\times 0.5}{2} + 8 \times \frac{4\times 1}{2}}{\frac{(4+8)\times 0.5}{2} + \frac{4\times 1}{2}} = 5.6$



Center of sums defuzzification method.

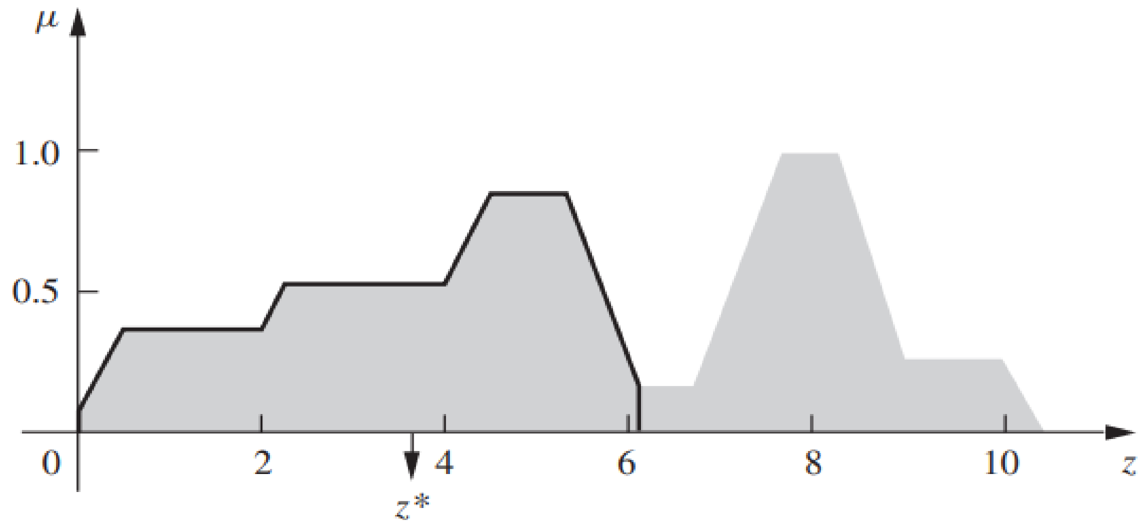
Defuzzification Methods

6. Center of Largest Area

- It is the center of gravity method but the centroid is computed for the largest convex sub-region.

- *Continuous form:* $z^* = \dfrac{\int \mu_{C_m}(z) z dz}{\int \mu_{C_m}(z) dz}$ where $\int$ denotes an algebraic integration, $\underset{\sim}{C}_m$ is the largest convex sub-region of the inferred output membership functions.

- *Discrete form:* $z^* = \dfrac{\sum_{z_i \in Z} \mu_{C_m}(z_i) z_i}{\sum_{z_i \in Z} \mu_{C_m}(z_i)}$ where $\sum$ denotes an algebraic sum.



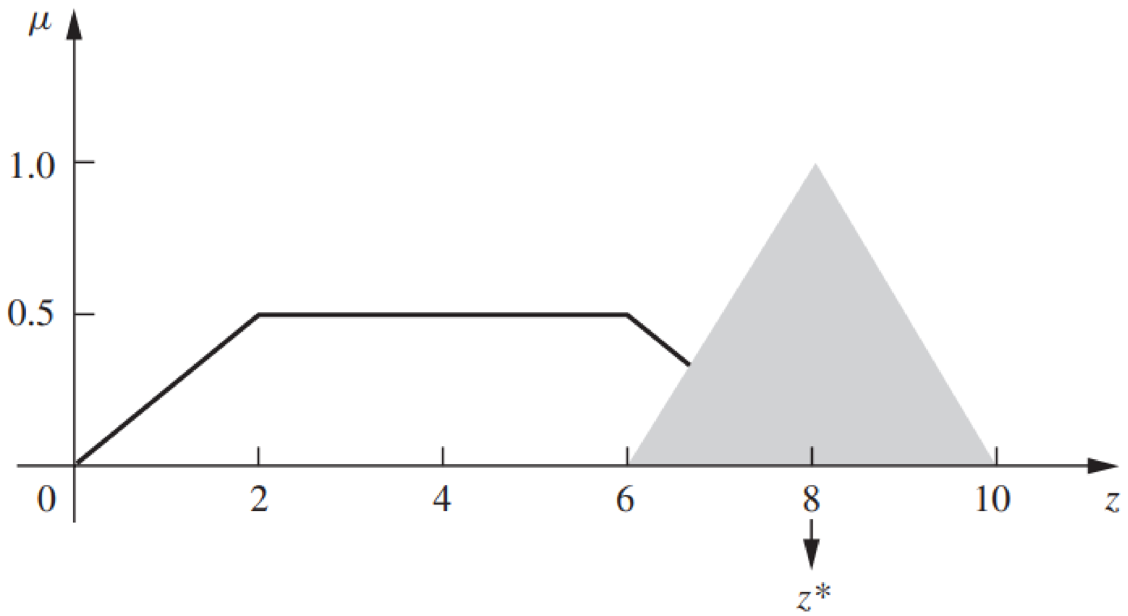
Center of largest area defuzzification method.

Defuzzification Methods

7. First (or last) of Maxima

- *The first of the maxima:* $z^* = \inf_{z \in Z}\{z \in Z \mid \mu_C(z) = \text{hgt}(\mu_C)\}$.
- *The last of maxima:* $z^* = \sup_{z \in Z}\{z \in Z \mid \mu_C(z) = \text{hgt}(\mu_C)\}$

where *inf* and *sup* stand for *infimum* and *supremum*, respectively.



First (or last) of maxima defuzzification method.

This method uses the overall output or union of all individual output fuzzy sets for determining the smallest value of the domain maximized membership

"SUP" is supremum i.e., least upper bound. The first of maximum is found.

"inf" is the infimum i.e., greatest lower bound . The last maximum is found

Defuzzification Methods

8. Lambda-cut method

Lambda-cut method is applicable to derive crisp value of a fuzzy set or relation.

- Thus Lambda-cut method for fuzzy set
- Lambda-cut method for fuzzy relation

- In many literature, Lambda-cut method is also alternatively termed as alpha-cut method.

Lambda-cut method

In this method a fuzzy set A is transformed into a crisp set A_λ for a given value of λ $(0 \leq \lambda \leq 1)$

In other-words, $A_\lambda = \{x | \mu_A(x) \geq \lambda\}$

$$A_1 = \{(x_1, 0.9), (x_2, 0.5), (x_3, 0.2), (x_4, 0.3)\}$$

Then $A_{0.6} = \{(x_1, 1), (x_2, 0), (x_3, 0), (x_4, 0)\} = \{x_1\}$

and

$$A_2 = \{(x_1, 0.1), (x_2, 0.5), (x_3, 0.8), (x_4, 0.7)\}$$

$$A_{0.2} = \{(x_1, 0), (x_2, 1), (x_3, 1), (x_4, 1)\} = \{x_2, x_3, x_4\}$$

Lambda-cut method

The Lambda-cut method for a fuzzy set can also be extended to fuzzy relation also.

Example: For a fuzzy relation R

$$R = \begin{bmatrix} 1 & 0.2 & 0.3 \\ 0.5 & 0.9 & 0.6 \\ 0.4 & 0.8 & 0.7 \end{bmatrix}$$

We are to find λ-cut relations for the following values of $\lambda = 0, 0.2, 0.9, 0.5$

$$R_0 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \text{ and } R_{0.2} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \text{ and}$$

$$R_{0.9} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \text{ and } R_{0.5} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

Application of Fuzzy Systems

Consumer products:

- washing machines
- microwave ovens
- rice cookers
- vacuum cleaners
- word translators
- Air conditioning

Systems:

- elevators
- train
- cranes
- automotive (engines, transmissions, brakes)
- traffic control

Software:

- medical diagnosis
- securities
- data compression

Advantages of Fuzzy system

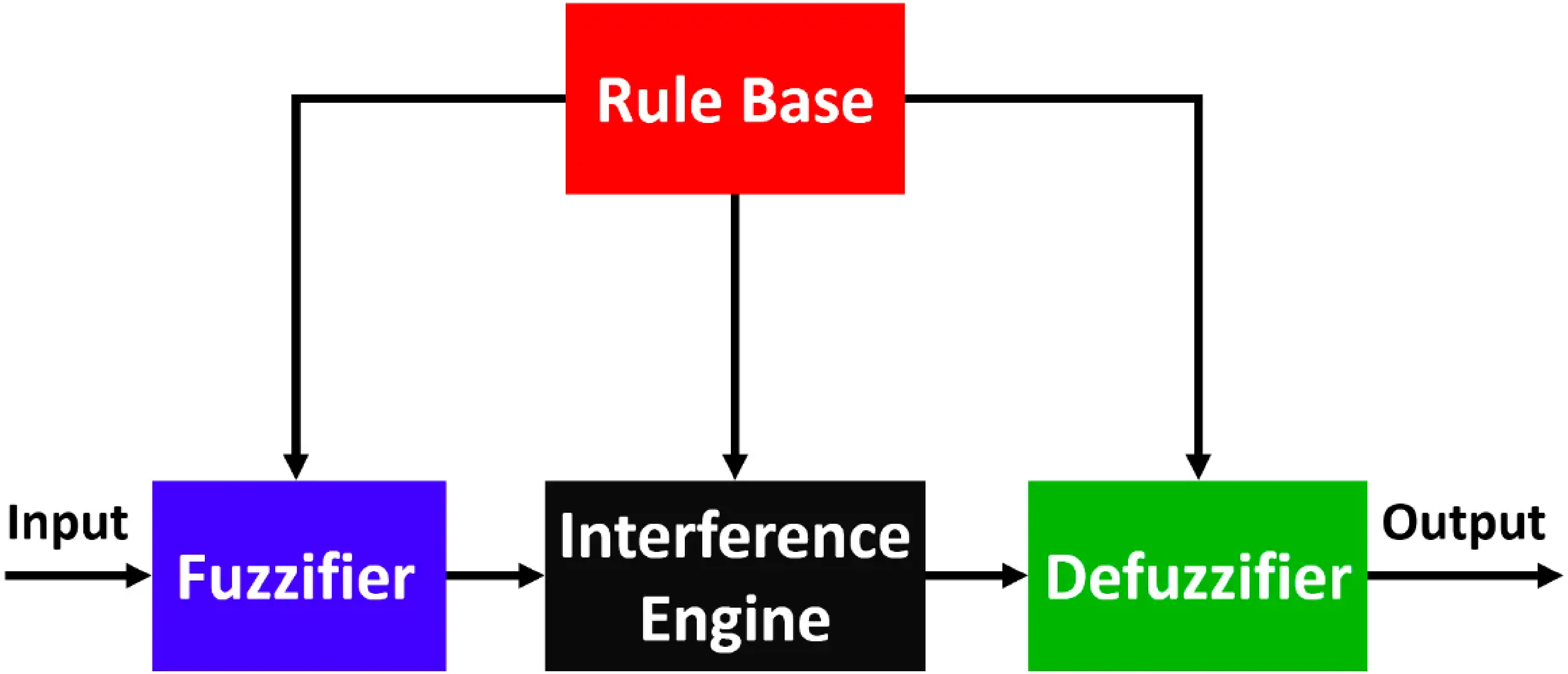
- Many complex problems cannot be solved by other systems which are easily solved by fuzzy logic.
- Fuzzy logic can give accurate outputs with imprecise data or inaccurate.
- Fuzzy logic is widely used for commercial and practical purposes.

Limitation of Fuzzy Systems

- Fuzzy systems lack the capability of machine learning as well as neural network-type pattern recognition
- Verification and validation of a fuzzy knowledge-based system require extensive testing with hardware
- Determining exact fuzzy rules and membership functions is a hard task
- Stability is an important concern for fuzzy control
- Setting up fuzzy logic itself is sometimes very difficult

The Architecture of Fuzzy Logic

Fuzzy Logic has Four main components. They are Fuzzification, Rule Base, Interference Engine, and Defuzzifier.



Four Components

1. **Fuzzification**: In fuzzification, the inputs from various input devices like sensors are converted into fuzzy sets. The inputs from the input devices are also termed crisp inputs.
2. **Rule Base:** Rule base as the name suggests contains the rules. The basics of this rule base are IF-THEN. The conditions are taken into consideration and the rule of IF is applied. If the condition meets the requirements of the given logical statement, then the rule is applied. Here, Rule Base is used by the system to determine the output.
3. **Interference Engine:** The interference engine is the main brain of fuzzy logic. The interference engine decides what to do or what output to produce for a given fuzzy input set. The output of the interference engine is also in terms of fuzzy sets.
4. **Defuzzification**: The fuzzy sets given by the interference engine as output is an inputs for the defuzzifier. Taking the fuzzy outputs and converting them to a single or crisp output value.

The END