UNIT III

Syllabus

- **Fuzzy Systems and Applications:**
- Fuzzy sets;
- fuzzy reasoning,
- fuzzy inference systems,
- fuzzy control,
- fuzzy clustering,
- applications of fuzzy systems.

Motivation

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

Fuzzy Systems

- A system becomes a fuzzy system when its operations are entirely or partially governed by fuzzy logic or are based on fuzzy sets.

Fuzzy system design steps

1. Knowledge acquisition
2. Translate into the language of fuzzy inference, composing fuzzy rules, and designing fuzzy variables
3. Developing the basic fuzzy inference algorithm, filling the gaps by appropriate selections, and designing heuristic elements
4. Simulation and testing

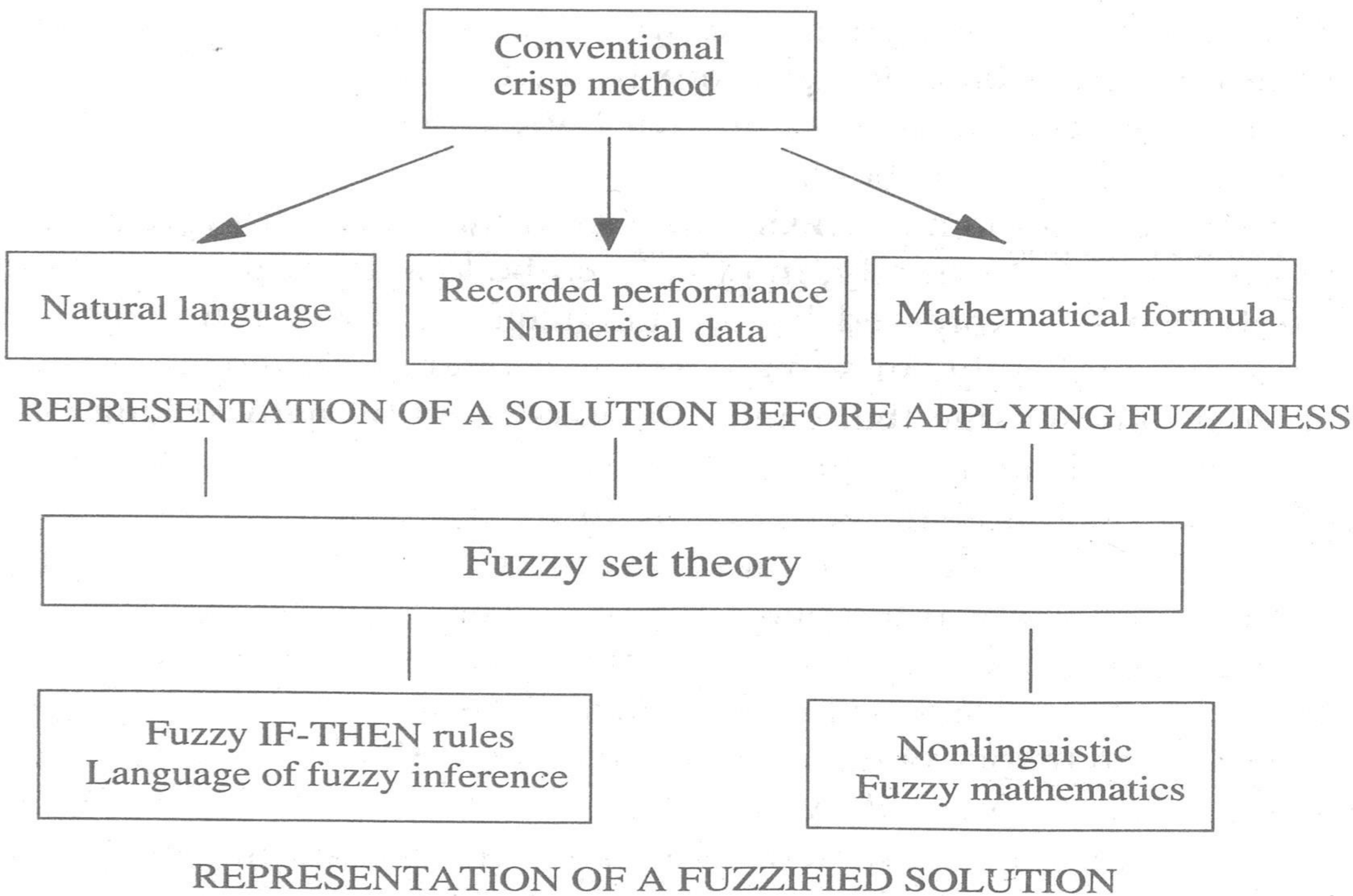
Knowledge acquisition

- Fuzzy system starts with obtaining the conventional method of solution.
- This is called the knowledge acquisition process.
- Conventional method of solutions are
  - Articulated expertise of the human operator, such as explaining how to ride a bike (natural language)
  - Recorded performance data (numerical data)
  - Closed-form mathematical formula

Translation

- The design challenge is to translate the knowledge given in one of these forms into fuzzy IF-THEN form.
- **Expertise articulated** in natural language is readily compatible with fuzzy IF-THEN rules.
- Knowledge in the form of **recorded performance** represent a more difficult case for translation into fuzzy IF-THEN form. There are approximate methods for translation from the numerical data domain into IF-THEN language.
- Translating a **mathematical expression** into fuzzy IF-THEN rules is also not very easy. A closed-form formula may contain knowledge that cannot be expressed by natural language. In such a case, the application of fuzzy set theory produces a fuzzy mathematical solution instead of a translation into fuzzy inference language. Examples are fuzzy arithmetic and fuzzy graph theory.

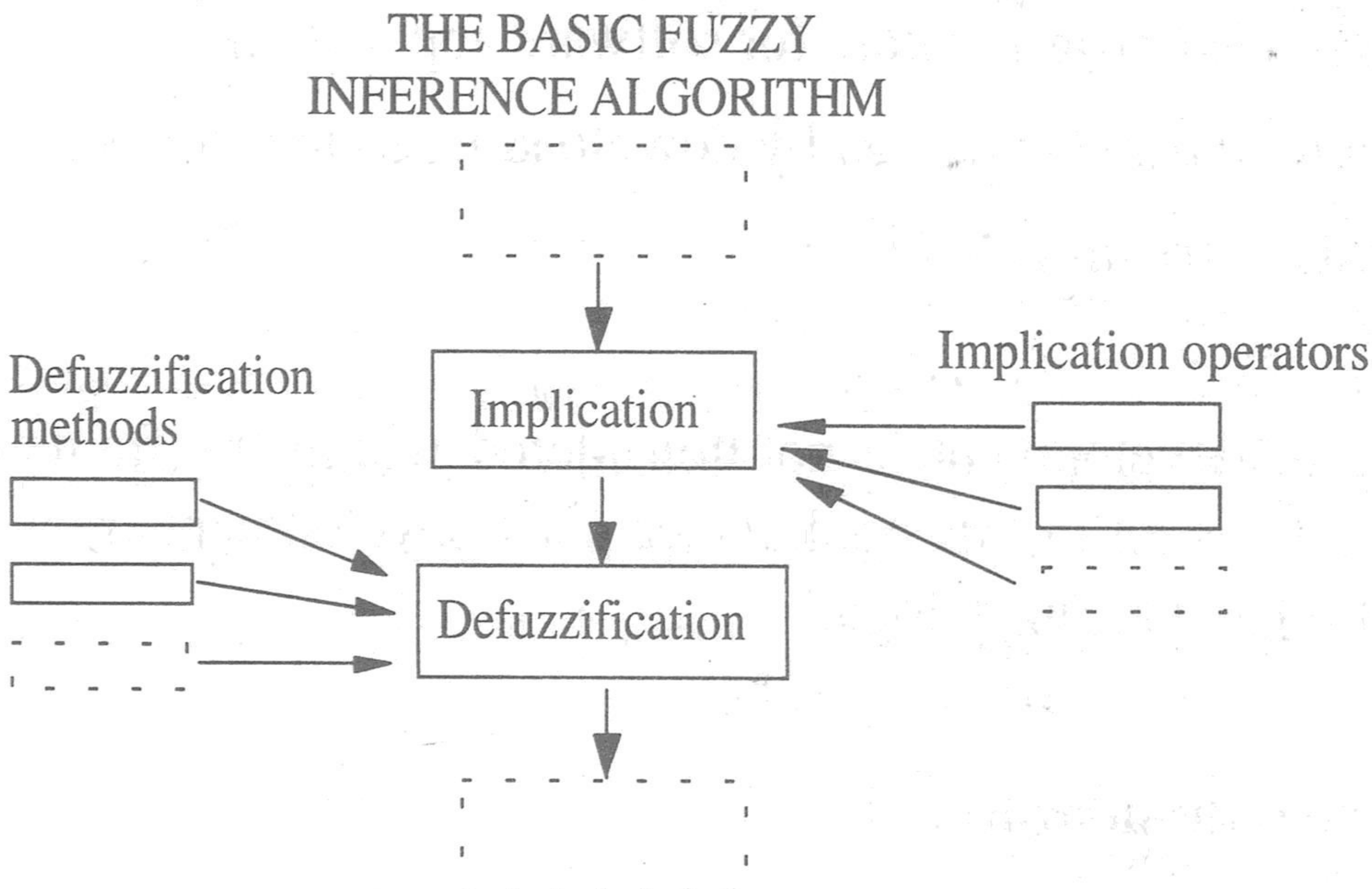
Converting a conventional method into a fuzzy method



Developing the basic fuzzy inference algorithm

- Basic fuzzy inference algorithm, which is a skeleton of work flow to be filled with designable elements, sometimes requiring selection from multiple options.

Fuzzy system design



Simulation and testing

- Simulation is defined as the process of creating a *model* of an existing or proposed *system* in order to identify and understand those factors which control the system and/or to predict (forecast) the future behavior of the system.

How are fuzzy systems implemented?

- The design involves special purpose software products that include design tools.
- Once a development is finalized, the fuzzy algorithm is converted into a computer program.
- Then the program is imported into the application environment.
- If this environment is a computer program or a software package, then fuzzy algorithms are embedded.
- If this environment is digital hardware such as VLSI chips, then fuzzy algorithms are downloaded (or burned) on digital platforms.
- If the hardware is an analog device, then the fuzzy algorithm is implemented by circuitry design.

How are fuzzy systems implemented

There are three forms of fuzzy system implementation:

- Embedded programming for software applications
- Digital hardware design by downloading source codes
- Analog circuitry design

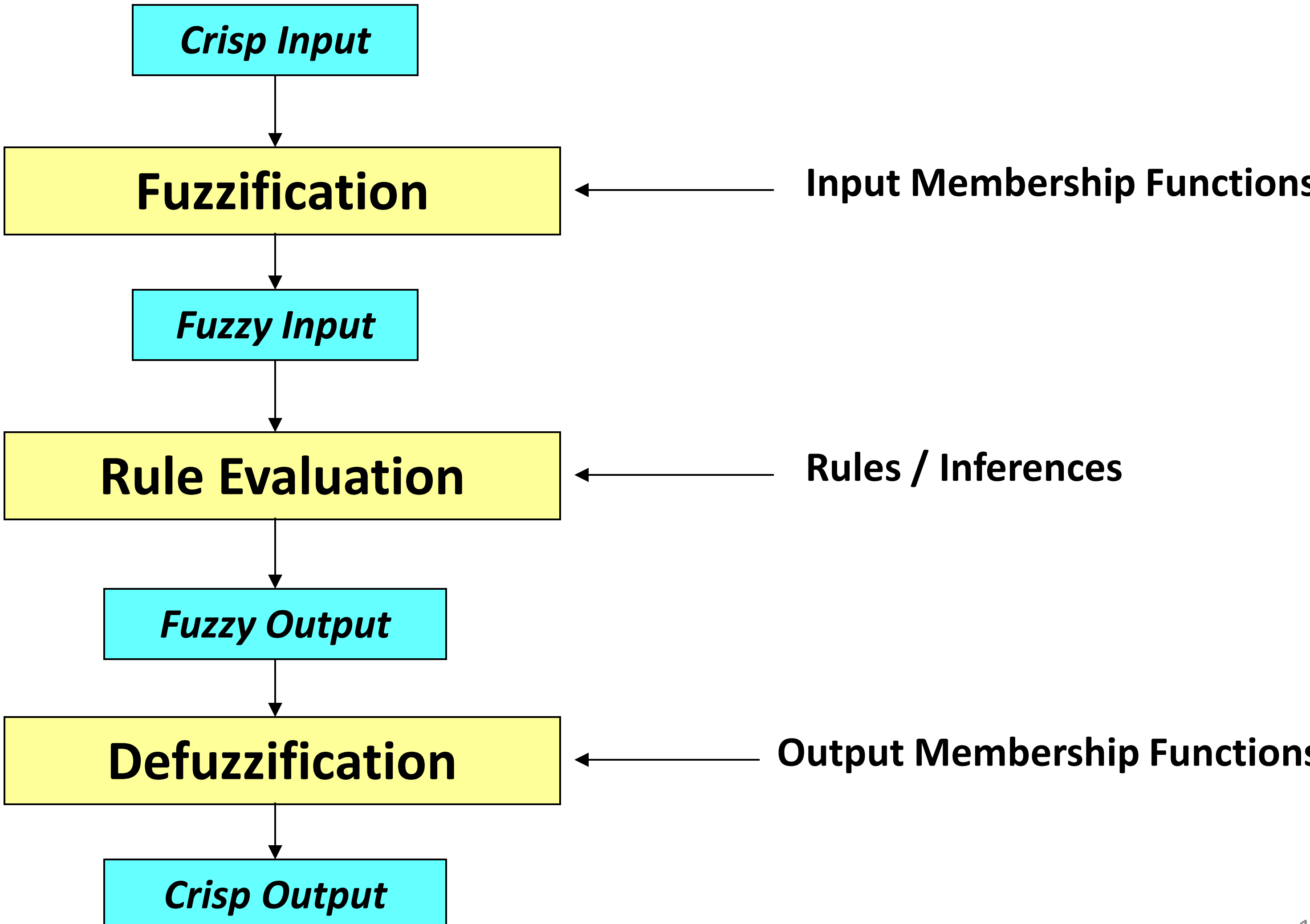
Who is the designer?

- Three different specialties are required to complete a working fuzzy system
  - An expert who knows the conventional non-fuzzy solution or method
  - A fuzzy logic specialist who can perform the design steps
  - A software/hardware specialist who can implement the design

What are the design tools?

- Fuzzy system design requires a computer-aided design environment in which some of the tedious tasks are automated or already designed.

Operation of Fuzzy System



Building Fuzzy Systems

- Fuzzification
- Inference
- Composition
- Defuzzification

Fuzzification

- Establishes the fact base of the fuzzy system.
- It identifies the input and output of the system, defines appropriate IF THEN rules, and uses raw data to derive a membership function.
- Consider an air conditioning system that determine the best circulation level by sampling temperature and moisture levels. The inputs are the current temperature and moisture level. The fuzzy system outputs the best air circulation level: “none”, “low”, or “high”. The following fuzzy rules are used:

1. If the room is hot, circulate the air a lot.
2. If the room is cool, do not circulate the air.
3. If the room is cool and moist, circulate the air slightly.

   – A knowledge engineer determines membership functions that map temperatures to fuzzy values and map moisture measurements to fuzzy values.

Inference

- Evaluates all rules and determines their truth values.
- If an input does not precisely correspond to an IF THEN rule, partial matching of the input data is used to interpolate an answer.
- Continuing the example, suppose that the system has measured temperature and moisture levels and mapped them to the fuzzy values of .7 and .1 respectively.
- The system now infers the truth of each fuzzy rule. To do this a simple method called MAX-MIN is used.
- This method sets the fuzzy value of the THEN clause to the fuzzy value of the IF clause.
- Thus, the method infers fuzzy values of 0.7, 0.1, and 0.1 for rules 1, 2, and 3 respectively.

Composition

- Combines all fuzzy conclusions obtained by inference into a single conclusion. Since different fuzzy rules might have different conclusions, consider all rules.
- Continuing the example, each inference suggests a different action
  - rule 1 suggests a "high" circulation level
  - rule 2 suggests turning off air circulation
  - rule 3 suggests a "low" circulation level.
- A simple MAX-MIN method of selection is used where the maximum fuzzy value of the inferences is used as the final conclusion. So, composition selects a fuzzy value of 0.7 since this was the highest fuzzy value associated with the inference conclusions.

Defuzzification

- Convert the fuzzy value obtained from composition into a “crisp” value.
- This process is often complex since the fuzzy set might not translate directly into a crisp value.
- Defuzzification is necessary, since controllers of physical systems require discrete signals.
- Continuing the example, composition outputs a fuzzy value of 0.7. This imprecise value is not directly useful since the air circulation levels are “none”, “low”, and “high”.
- The defuzzification process converts the fuzzy output of 0.7 into one of the air circulation levels. In this case it is clear that a fuzzy output of 0.7 indicates that the circulation should be set to “high”.

Fuzzy Control

- Let us assume that this fuzzy logic controller (FLC) control a hot water heater.
- The hot water heater has a knob, heatknob (0-10) on it to control the heating element power, the higher the value, the hotter it gets, with a value of 0 indicating the heating element is turned off.
- There are two sensors in the hot water heater, one to tell you the temperature of the water(tempsense) , which various from 0 to 125 C, and the other to tell you the level of the water in the tank(levelsense), which various from 0=empty to 10=full.
- Assume that there is an automatic flow control that determines how much cold water flows into the tank from the main water supply; when ever the level of the water gets below 40, the flow control turns on, and turns off when the level of the water gets above 95.

Step 1: defining inputs and outputs for the FLC

- The range of values that inputs and outputs may take is called the universe of discourse.
- We need to define the universe of discourse for all of the inputs and outputs of the FLC, which are all crisp values

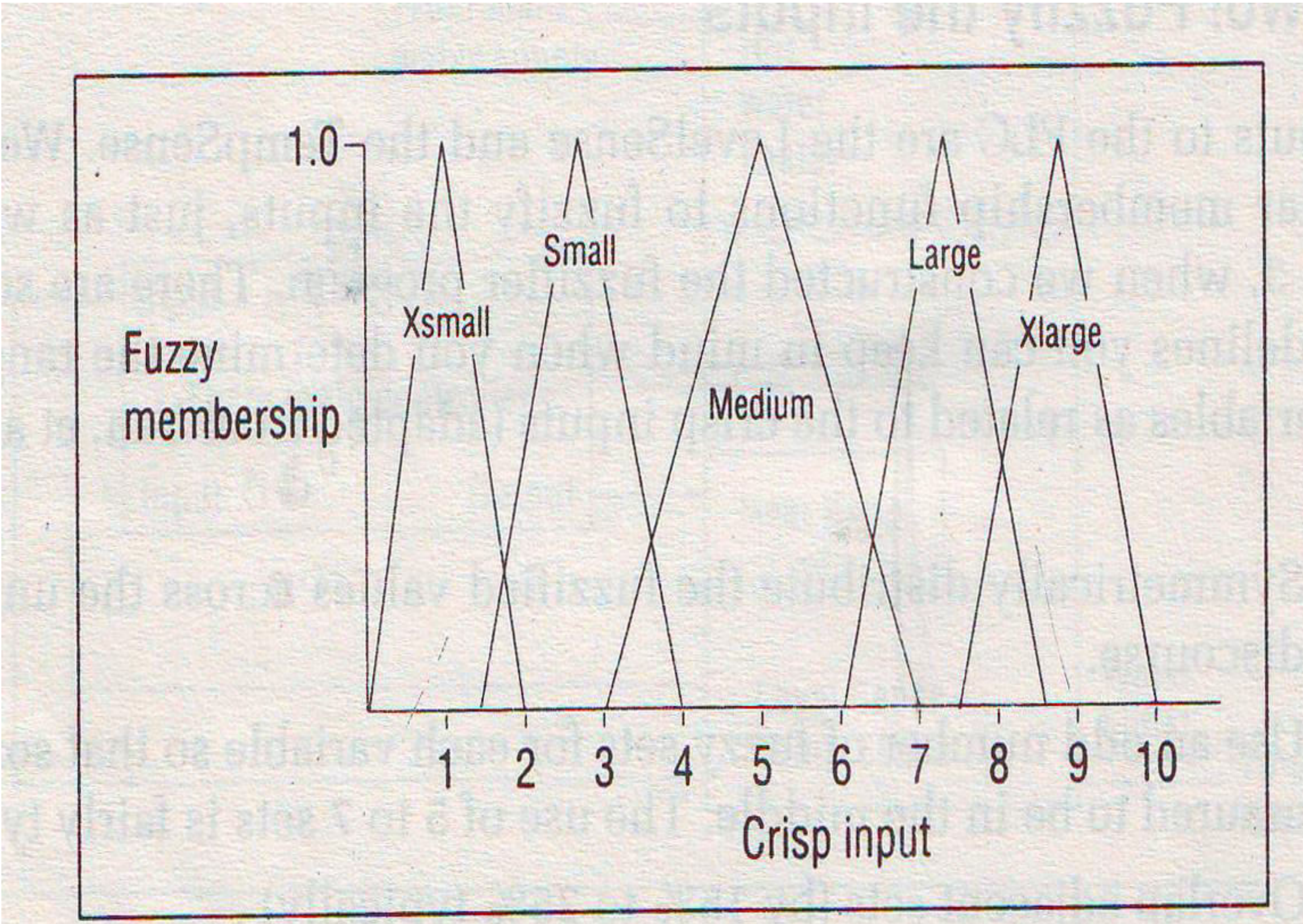
Name	input/output	minimum value	maximum value
LevelSense	I	0	10
HeatKnob	O	0	10
TempSense	I	0	125

Step 2: Fuzzify the Inputs

- The inputs to the FLC are the LevelSense and the TempSense.
- Fuzzy variables ranges for LevelSense

Crisp Input Range	Fuzzy Variables
0-2	Xsmall
1.5-4	Small
3-7	Medium
6-8.5	Large
7.5-10	XLarge

Fuzzy Membership functions for LevelSense

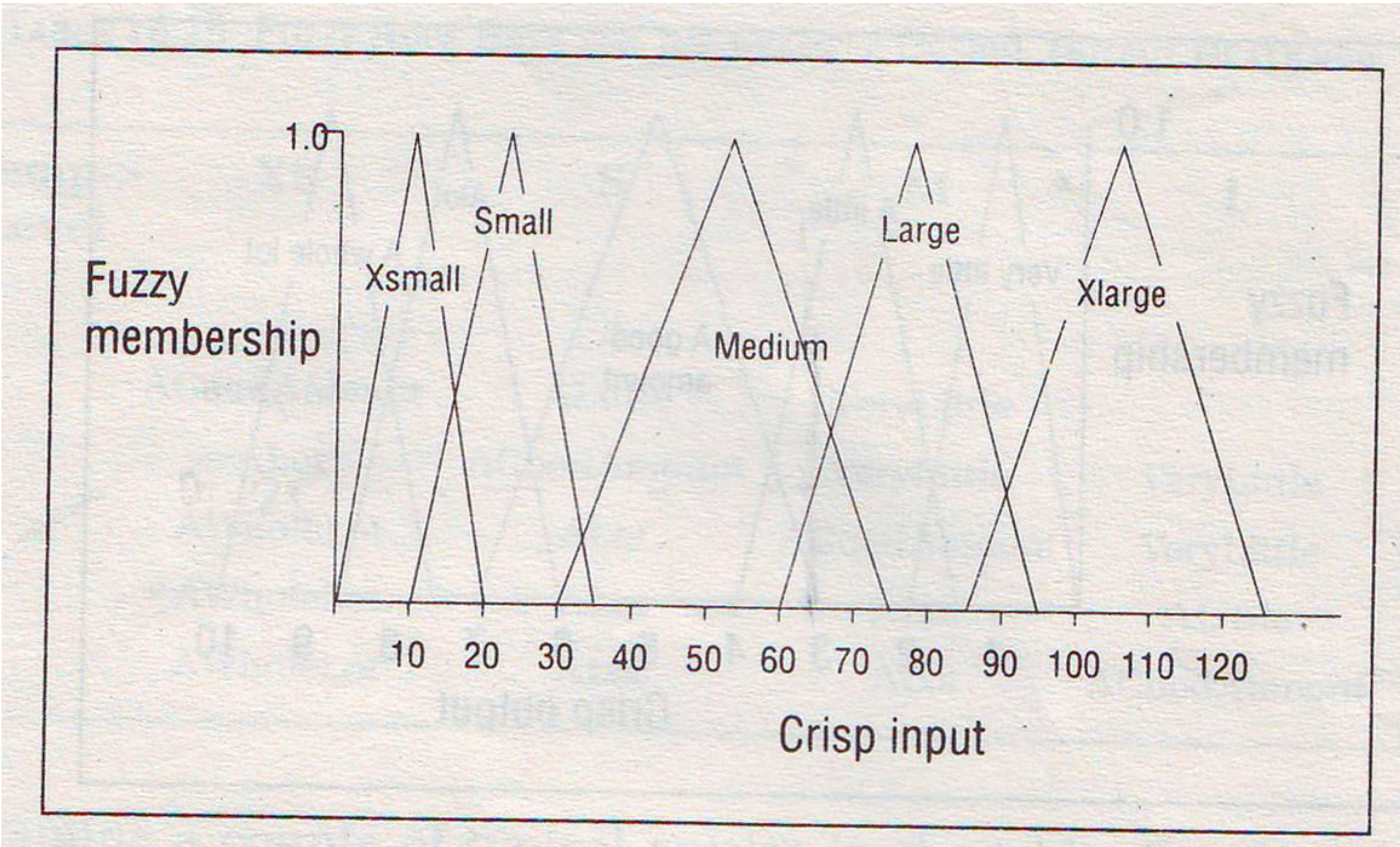


Step 2: Fuzzify the Inputs

Fuzzy variables ranges for TempSense

Crisp Input Range	Fuzzy Variables
0-20	Xsmall
10-35	Small
30-75	Medium
60-95	Large
85-125	XLarge

Fuzzy Membership functions for TempSense

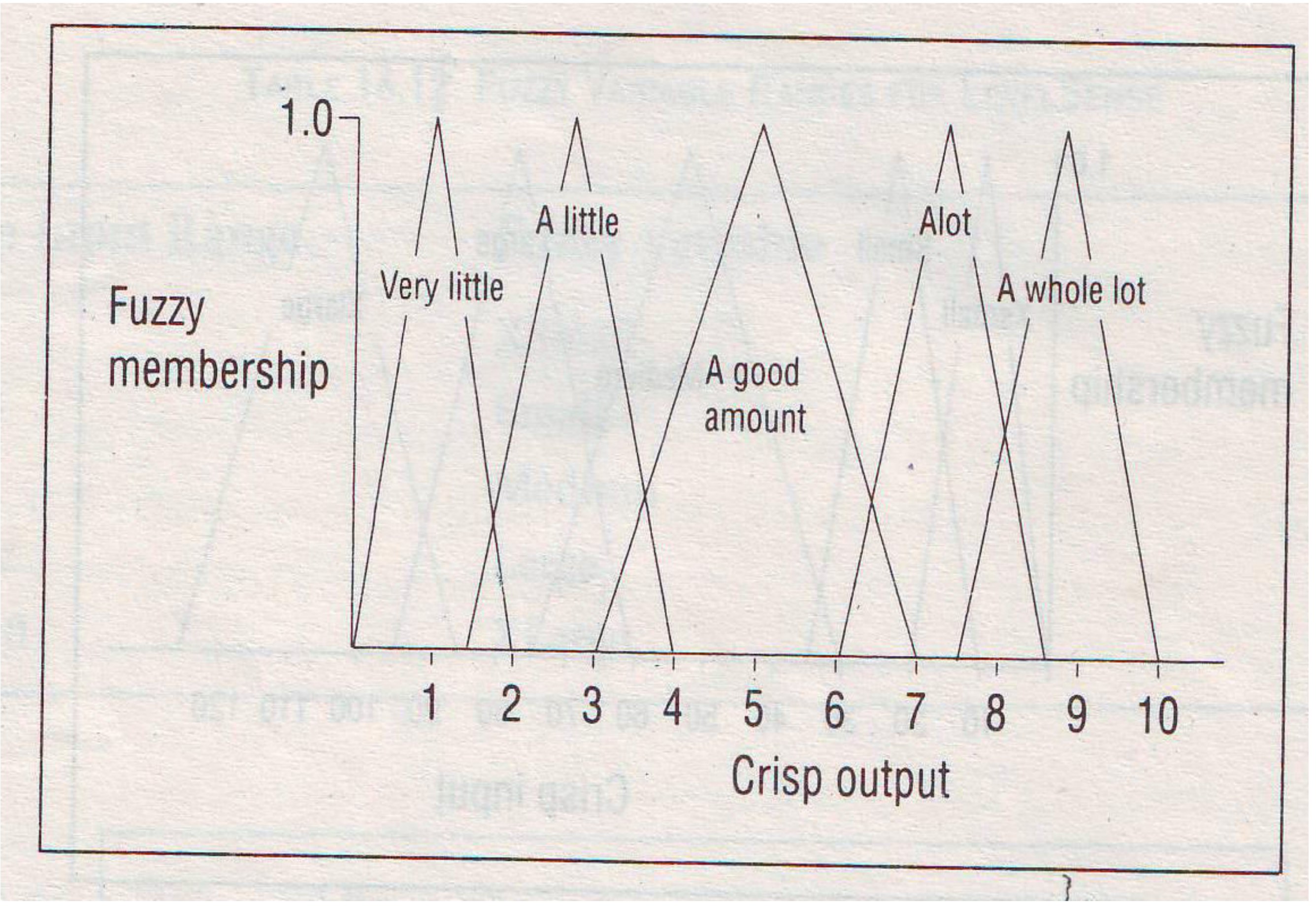


Step 3: Set up Fuzzy Membership Functions for the outputs

- We have just one output, which is the HeatKnob.

Crisp Output Range	Fuzzy Variables
0-2	**VeryLittle**
1.5-4	**Alittle**
3-7	**AGoodAmount**
6-8.5	**A lot**
7.5-10	**AWholeLot**

Fuzzy Membership functions for HeatKnob



Step 4: Create a Fuzzy Rule Base

- Now we have the inputs and the output defined in terms of fuzzy variables.
- Now we need to specify what actions to take under what conditions.
- That is we need to construct a set of rules that describe the operation of the FLC.
- These rules take the form of IF-THEN rules and can be obtained from a human expert.

Step 4: Create a Fuzzy Rule Base

SenseTemp	XS	S	M	L	XL
SenseLevel					
XS	AGoodAmount	ALittle	VeryLittle		
S	ALot	AGoodAmount	VeryLittle	VeryLittle	
M	AWholeLot	ALot	AGoodAmount	VeryLittle	
L	AWholeLot	ALot	ALot	ALittle	
XL	AWholeLot	ALot	ALot	AGoodAmount	

Creating IF-THEN Rules

- We can now translate the table entries into IF-THEN rules.

1. IF SenseTemp IS XSmall AND SenseLevel IS Xsmall THEN SET HeatKnob TO AGoodAmount
2. IF SenseTemp IS XSmall AND SenseLevel IS Xsmall THEN SET HeatKnob TO ALot
3. IF SenseTemp IS XSmall AND SenseLevel IS Medium THEN SET HeatKnob TO AWholeLot
4. IF SenseTemp IS XSmall AND SenseLevel IS Large THEN SET HeatKnob TO AWholeLot
5. IF SenseTemp IS XSmall AND SenseLevel IS XLarge THEN SET HeatKnob TO AWholeLot

Creating IF-THEN Rules

6. IF SenseTemp IS Small AND SenseLevel IS Xsmall THEN SET HeatKnob TO ALittle
7. IF SenseTemp IS Small AND SenseLevel IS Small THEN SET HeatKnob TO AGoodAmount
8. IF SenseTemp IS Small AND SenseLevel IS Medium THEN SET HeatKnob TO ALot
9. IF SenseTemp IS Small AND SenseLevel IS Large THEN SET HeatKnob TO ALot
10. IF SenseTemp IS Small AND SenseLevel IS XLarge THEN SET HeatKnob TO ALot
11. IF SenseTemp IS Medium AND SenseLevel IS XSmall THEN SET HeatKnob TO VeryLittle
12. IF SenseTemp IS Medium AND SenseLevel IS small THEN SET HeatKnob TO VeryLittle

Creating IF-THEN Rules

13. IF SenseTemp IS Medium AND SenseLevel IS Medium THEN SET HeatKnob TO AGoodAmount
14. IF SenseTemp IS Medium AND SenseLevel IS Large THEN SET HeatKnob TO ALot
15. IF SenseTemp IS Medium AND SenseLevel IS XLarge THEN SET HeatKnob TO ALot
16. IF SenseTemp IS Large AND SenseLevel IS small THEN SET HeatKnob TO VeryLittle
17. IF SenseTemp IS Large AND SenseLevel IS Medium THEN SET HeatKnob TO VeryLittle
18. IF SenseTemp IS Large AND SenseLevel IS Large THEN SET HeatKnob TO ALittle
19. IF SenseTemp IS Large AND SenseLevel IS XLarge THEN SET HeatKnob TO AGoodAmount

Step 5: Defuzzify the Outputs

- In order to control the HeatKnob, we need to obtain a crisp dial setting.
- So far, we have several of the IF–THEN rules of the fuzzy rule base firing at once, because the inputs have been fuzzified.
- How do we arrive at a single crisp output number ?
- There are actually several different strategies for this; we will consider two of the most common, the center of area (COA) or centroid method, and the fuzzy Or method.
- The easiest way to understand the process is with an example.

Example

- Assume that at a particular point in time, LevelSense = 7.0 and TempSense = 65.
- These are the crisp inputs directly from the sensors. With fuzzification , assume that you get the following fuzzy memberships:

crisp input — LevelSense = 7.0
fuzzy outputs with membership values -
Medium: 0.4
Large: 0.6
all others : 0.0
crisp input — TempSense=65
fuzzy outputs with membership values -
Medium: 0.75
Large: 0.25
all others: 0.0

Example

This results in four rules firing:

1. TempSense = Medium (0.75) AND LevelSense = Medium (0.4)

2. TempSense = Large (0.25) AND LevelSense = Medium (0.4)

3. TempSense = Medium (0.75) AND LevelSense = Large (0.6)

4. TempSense = Large (0.25) AND LevelSense = Large (0.6)

- First you must determine, for each of the AND clauses in the IF–THEN rules, what the output should be.
- This is done by the conjunction or minimum operator. So for each of these rules you have the following firing strengths:

Example

1. (0.75) ^ (0.4) = 0.4
2. (0.25) ^ (0.4) = 0.25
3. (0.75) ^ (0.6) = 0.6
4. (0.25) ^ (0.6) = 0.25

By using the fuzzy rule base and the strengths assigned previously, we find the rules recommend the following output values (with strengths) for HeatKnob:

1. AGoodAmount (0.4)
2. VeryLittle (0.25)
3. ALot (0.6)
4. ALittle (0.25)

Example

- Now we must combine the recommendations to arrive at a single crisp value.
- Here we use a *disjunction or maximum* operator to combine the values.
- We obtain the following:
- $(0.4) \vee (0.25) \vee (0.6) \vee (0.25) = 0.6$
- The crisp output value for HeatKnob would then be this membership value multiplied by the range of the output variable, or (0.6) (10-0) = 6.0.

Advantages of Fuzzy Logic Controllers

- Relates input to output in linguistic terms, easily understood
- Allows for rapid prototyping because the system designer doesn’t need to know everything about the system before starting
- Cheaper because they are easier to design
- Increased robustness
- Simplify knowledge acquisition and representation
- A few rules encompass great complexity
- Can achieve less overshoot and oscillation
- Can achieve steady state in a shorter time interval

Disadvantages of Fuzzy Logic Controllers

- Hard to develop a model from a fuzzy system
- Require more fine tuning and simulation before operational
- Have a stigma associated with the word fuzzy; engineers and most other people are used to crispness and shy away from fuzzy control and fuzzy decision making

Linguistic Variables

Linguistic Variables

- A **variable** is a symbolic name given to some known or unknown quantity or information.
- A numerical variables takes numerical values: *Age = 65*
- A linguistic variables takes linguistic values: *Age is old*
- A linguistic values 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, ...}

Linguistic Variables

- Linguistic variable is "a variable whose values are words or sentences in a natural or artificial language".
- 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.
- Each linguistic variable may be assigned one or more linguistic values, which are in turn connected to a numeric value through the mechanism of membership functions.

Linguistic Variables

- In fuzzy expert systems, **linguistic variables are used in fuzzy rules**. For example:

IF	wind	is strong
THEN	sailing	is good
IF	project_duration	is long
THEN	completion_risk	is high
IF	speed	is slow
THEN	stopping_distance	is short

Example

if temperature is cold and oil is cheap

then heating is high

Example

$$\text{if } \underbrace{\overbrace{\text{temperature}}^{\text{Linguistic Variable}} \text{ is } \overbrace{\text{cold}}^{\text{Linguistic Value}}}_{\mu_{\text{cold}}} \text{ and } \underbrace{\overbrace{\text{oil}}^{\text{Linguistic Variable}} \text{ is } \overbrace{\text{cheap}}^{\text{Linguistic Value}}}_{\mu_{\text{cheap}}}$$

$$\text{then } \overbrace{\underbrace{\text{heating}}_{\text{Linguistic Variable}} \text{ is } \underbrace{\text{high}}_{\text{Linguistic Value}}}^{\mu_{\text{high}}}$$

Definition [Zadeh 1973]

A linguistic variable is characterized by a quintuple

$$\left(x, T(x), U, G, M\right)$$

Name

Term Set

Universe

Syntactic Rule

Semantic Rule

Example

A linguistic variable is characterized by a quintuple

$$\left(x, T(x), U, G, M\right)$$

age

$$G(\text{age}) = \left\{ \begin{array}{l} \text{old, very old, not so old,} \\ \text{more or less young,} \\ \text{quite young, very young} \end{array} \right\}$$

[0, 100]

Example semantic rule:

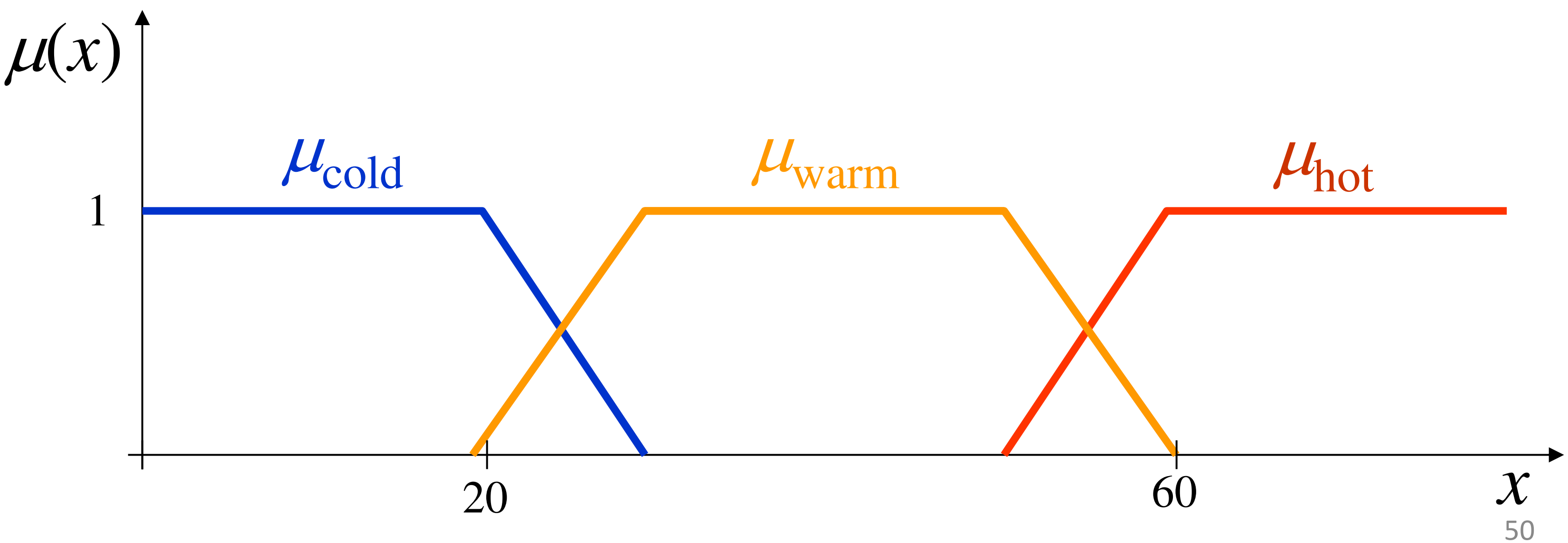
$$M(\text{old}) = \left\{ \left(u, \mu_{\text{old}}(u)\right) \middle| u \in [0,100] \right\}$$

$$\mu_{\text{old}}(u) = \begin{cases} 0 & u \in [0,50] \\ \left[1 + \left(\dfrac{u-50}{5}\right)^{-2}\right]^{-1} & u \in [50,100] \end{cases}$$

Example

Linguistic Variable : temperature

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



Linguistic Variables and Hedges

- The range of possible values of a linguistic variable represents the universe of discourse of that variable. For example, the universe of discourse of the linguistic variable *speed* might have the range between 0 and 220 km/h and may include such fuzzy subsets as *very slow*, *slow*, *medium*, *fast*, and *very fast*.
- A linguistic variable carries with it the concept of fuzzy set qualifiers, called **hedges**.
- Hedges are terms that modify the shape of fuzzy sets. They include adverbs such as *very*, *somewhat*, *quite*, *more or less* and *slightly*.

Operations on Linguistic Values

Concentration: $\Rightarrow$ $CON(A) = A^2$

Dilation: $\Rightarrow$ $DIL(A) = A^{0.5}$

Solve

- Let fuzzy set be A={(1,0),(2,0.1),(3,0.2),(4,0.5),(5,0.3),(6,0.1),(7,0.0),(8,0.0),(9,0.0),(10,1)}
- CON(A)=(2,0.01),(3,0.04),(4,0.25),(5,0.09),(6,0.01),(10,1)
- DIL(A)= (2,0.31),(3,0.44),(4,0.70),(5,0.54),(6,0.31),(10,1)

What is a fuzzy rule?

•Fuzzy rules are useful for modeling human thinking, perception and judgment.

•A fuzzy rule can be defined as a conditional statement in the form:

IF x is A THEN y is B

•where x and y are linguistic variables; and A and B are linguistic values determined by fuzzy sets on the universe of discourses X and Y, respectively.

•“x is A” is called *antecedent* and “y is B” is called *consequent*.

What is the difference between classical and fuzzy rules?

A classical IF-THEN rule uses binary logic, for example,

Rule: 1
IF speed is > 100
THEN stopping_distance is long

Rule: 2
IF speed is < 40
THEN stopping_distance is short

- The variable speed can have any numerical value between 0 and 220 km/h, but the linguistic variable stopping_distance can take either value long or short.
- In other words, classical rules are expressed in the black-and-white language of Boolean logic.

We can also represent the stopping distance rules in a fuzzy form:

Rule: 1
IF speed is fast
THEN stopping_distance is long

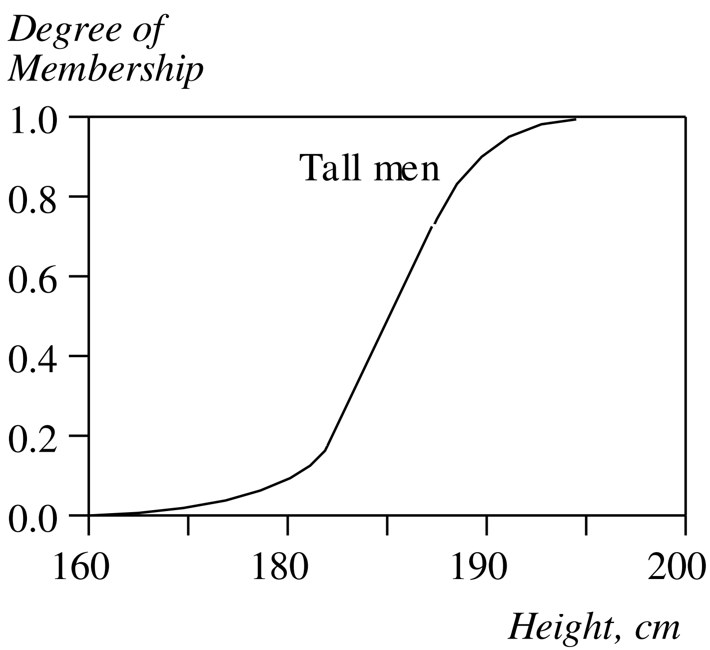
Rule: 2
IF speed is slow
THEN stopping_distance is short

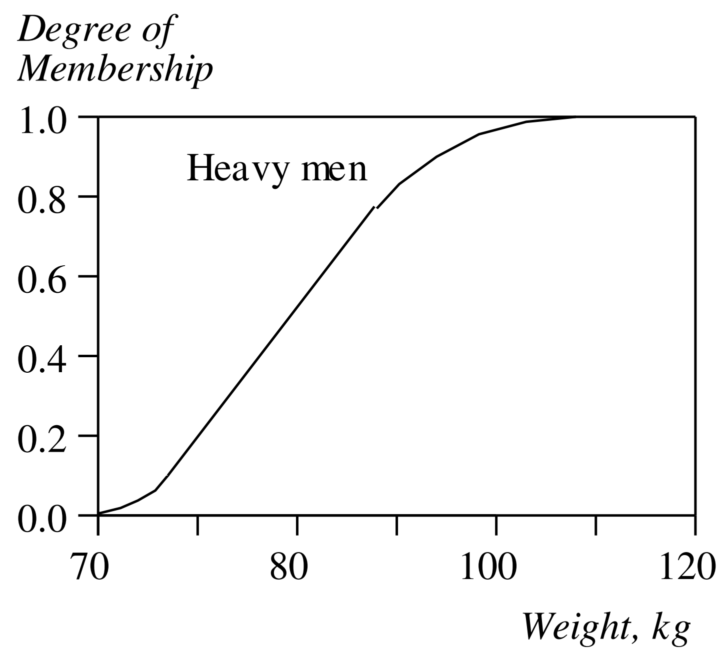
- In fuzzy rules, the linguistic variable speed also has the range (the universe of discourse) between 0 and 220 km/h, but this range includes fuzzy sets, such as slow, medium and fast.
- The universe of discourse of the linguistic variable stopping distance can be between 0 and 300 m and may include such fuzzy sets as short, medium and long.

Firing Fuzzy Rules

- These fuzzy sets provide the basis for a weight estimation model. The model is based on a relationship between a man's height and his weight:

  IF height is tall THEN weight is heavy





Firing Fuzzy Rules

- A fuzzy rule can have multiple antecedents, for example:

IF project_duration is long AND project_staffing is large AND project_funding is inadequate
THEN risk is high

IF service is excellent OR food is delicious
THEN tip is generous

- The consequent of a fuzzy rule can also include multiple parts, for instance:

IF temperature is hot
THEN hot_water is reduced;
cold_water is increased

Fuzzy Proposition

Proposition

- Propositions are elementary atomic sentences(we shall also use the term formulas or well-formed formulas in place of sentences).
- Propositions may be either true or false but may take on no other value.
- There are two kinds of proposition
  - Simple
  - compound
- Some examples of simple propositions are
  - It is raining.
  - My car is painted silver.
  - John and sue have five children.
  - Snow is white.
  - People live on the moon.

Proposition continued….

- Compound propositions are formed from atomic formulas using the logical connectives not or if ………then, and if and only if.
- For example, the following are compound formulas.
  - It is raining and the wind is blowing.
  - The moon is made of green cheese or it is not.
  - If you study hard you will be rewarded.
  - The sum of 10 and 20 is not 50.

Logic

- One of the prime activity of human intelligence is reasoning.
- The activity of reasoning involves construction, organization and manipulation of statements to arrive at new conclusions.
- Logic is the study of the methods and principles of reasoning in all its possible forms.

Propositional logic

- Propositional logic is a representational language that makes the assumption that the world can be represented solely in terms of propositions that are true or false.
- One of the main concerns of propositional logic is the study of rules by which new logic variables can be produced as functions of some given logic variables.

Fuzzy Proposition

- Fuzzy Proposition – Propositions that include fuzzy predicates. For example,
  - *It will be sunny today.*
  - *Dow Jones is closed higher yesterday.*
- Canonical Form – (Unconditional fuzzy proposition)

  $x \quad is \quad A$

  A (a fuzzy set) is a fuzzy predicate called the *fuzzy variable* or the *linguistic variable*. The values of a linguistic variable are words or sentences in a natural or synthetic language.

Fuzzy Proposition

- Note that a linguistic variable is a fuzzy (sub)set defined on a *Universe of discourse*. For example,

  Janet is young

  implies the *AGE* of Janet is Young. Here, Young is a fuzzy set defined on the axis "Age".
- Other fuzzy sets may be defined on the same universe include "Old", "Mid-age", etc.
- Age is a property of "Janet", and Young is a specific subset of "Age".

Syntax for Propositional logic

- Propositional logic consists of propositional symbols such as P,Q,R,S and logic connectives given below.

$\wedge$ (and)	This is called *conjunction* and is used for constructing a sentence like $P \wedge Q$
$\vee$ (or)	This is called *disjunction* and is used for constructing a sentence like $P \vee Q$
$\neg$ (negation)	This is called *not* and is used in constructing sentence like $\neg P$
$\Rightarrow$ (implication)	This is used in constructing sentences like $P \Rightarrow Q$ and is equivalent to $\neg P \vee Q$
$\Leftrightarrow$ (equivalent)	These are used in constructing a sentence like $P \Leftrightarrow Q$ which implies P is equivalent to Q

Example

Write the syntax for propositional logic
If the road is closed, then the traffic is blocked.

“the road is closed” is represented by a proposition, P .
“then the traffic is blocked” is represented by a proposition, Q .
The sentence is represented as

$$P \Rightarrow Q.$$

Semantics for Propositional logic

The semantics or meaning of a sentence is just the value true or false. The terms used for semantics of a language are given below.

Valid	A sentence is valid if it is true for all interpretations. Valid sentences are also called tautologies.
Model	An interpretation of a formula or sentence under which the formula is true is called a model of that formula.
Unsatisfiable (contradiction)	It is said to be unsatisfiable if it is false for every interpretation.
Satisfiable	It is said to be satisfiable if it is true for some interpretation.
Equivalence	Two sentences are equivalent if they have the same truth value under every interpretation.

Truth tables for logical connectives

A	B	~A	~B	AvB	A&B	A->B	A<->B
T	T	F	F	T	T	T	T
F	T	T	F	T	F	T	F
T	F	F	T	T	F	F	F
F	F	T	T	F	F	T	T

EXAMPLE

- Let A be a proposition- **the machine is defective**

  and B be a proposition- **the production is less**
- The implication is written as,

If the machine is defective THEN production is less.

- Now we analyze all the combination to find whether the truth table is correct or not.

Solution

I. In the first condition, we have that if A is true and B is also true then A->B is also true, i.e if machine is defective then production is less which is true.

II. Secondly, if A is false and B is also true then A->B is true, i.e if machine is not defective then production is less. Of course it may be true because production can be less because of many reasons.

III. If A is true and B is false then A->B is false, i.e if machine is defective then production is not less. Of course it is false, as production will directly get affected by machine performance.

IV. If A is false and B is also false then A->B is true, i.e if machine is not defective then production is not less. No doubt it is true.

Show that

$P \rightarrow Q$ is equivalent to $\sim P \vee Q$

Show that

P <-> Q is equivalent to

$$(P \rightarrow Q) \,\&\, (Q \rightarrow P)$$

Show that

$P \to Q$ is equivalent to $\sim P \vee (P \wedge Q)$

Truth table for equivalent sentences

P	Q	¬P	(¬P ∨ Q)	(P → Q)	(Q → P)	(P → Q) & (Q → P)
true	true	false	true	true	true	true
true	false	false	false	false	true	false
false	true	true	true	true	false	false
false	false	true	true	true	true	true

Some Equivalence Laws

Idempotency	$P \vee P = P$ $P \& P = P$
Associativity	$(P \vee Q) \vee R = P \vee (Q \vee R)$ $(P \& Q) \& R = P \& (Q \& R)$
Commutativity	$P \vee Q = Q \vee P$ $P \& Q = Q \& P$ $P \leftrightarrow Q = Q \leftrightarrow P$
Distributivity	$P \& (Q \vee R) = (P \& Q) \vee (P \& R)$ $P \vee (Q \& R) = (P \vee Q) \& (P \vee R)$
De Morgan's laws	$\sim(P \vee Q) = \sim P \& \sim Q$ $\sim(P \& Q) = \sim P \vee \sim Q$
Conditional elimination	$P \rightarrow Q = \sim P \vee Q$
Bi-conditional elimination	$P \leftrightarrow Q = (P \rightarrow Q) \& (Q \rightarrow P)$

Question 1

- Determine whether each of the following sentences is (a) Satisfiable (b) Contradictory and (c) Valid.

S1: (P & Q) $\vee$ ~(P $\vee$ Q)

S2: (P $\vee$ Q)-> (P & Q)

S3: (P & Q) -> (P $\vee$ ~Q)

S4: (P $\vee$ Q) & (P $\vee$ ~Q) $\vee$ P

S5: P -> Q->~P

S6: P $\vee$ Q & ~P $\vee$ ~Q &P

S1

P	Q	P & Q	P ∨ Q	~(P ∨ Q)	(P & Q) ∨ ~(P ∨ Q)
T	T	T	T	F	T
F	T	F	T	F	F
T	F	F	T	F	F
F	F	F	F	T	T

S1: (P & Q) ∨ ~(P ∨ Q) satisfiable

S2

P	Q	$P \vee Q$	P & Q	$(P \vee Q)$->(P & Q)
T	T	T	T	T
F	T	T	F	F
T	F	T	F	F
F	F	F	F	T

S2: $(P \vee Q)$-> (P & Q) satisfiable

S3

P	Q	~Q	P & Q	P ∨ ~Q	(P & Q) -> (P ∨ ~Q)
T	T	F	T	T	T
F	T	F	F	F	T
T	F	T	F	T	T
F	F	T	F	T	T

S3: (P & Q) -> (P ∨ ~Q) valid

S4

P	Q	$P \vee Q$	$P \vee \sim Q$	$(P \vee Q) \& (P \vee \sim Q)$	$(P \vee Q) \& (P \vee \sim Q) \vee P$
T	T	T	T	T	T
F	T	T	F	F	F
T	F	T	T	T	T
F	F	F	T	F	F

S4: $(P \vee Q)$ & $(P \vee \sim Q) \vee P$ satisfiable

S5

P	Q	~P	P->Q	P -> Q->~P
T	T	F	T	F
F	T	T	T	T
T	F	F	F	T
F	F	T	T	T

S5: P -> Q->~P satisfiable

S6

P	Q	~P	~Q	P ∨ Q	(~P ∨ ~Q)	(P ∨ Q) & (~P ∨ ~Q)	(P ∨ Q) & (~P ∨ ~Q)& P
T	T	F	F	T	F	F	F
F	T	T	F	T	T	T	F
T	F	F	T	T	T	T	T
F	F	T	T	F	T	F	F

S6: P ∨ Q & ~P ∨ ~Q &P satisfiable

Question 2

- Construct the truth table

(P <-> (Q ∧ R)) -> (~R-> ~Q)

Solution 2

P	Q	R	Q ∧ R	P<->(Q ∧ R)	~R	~Q	~R -> ~Q	(P <-> (Q ∧ R)) -> (~R-> ~Q)
T	T	T	T	T	F	F	T	T
T	T	F	F	F	T	F	F	T
T	F	T	F	F	F	T	T	T
T	F	F	F	F	T	T	T	T
F	T	T	T	F	F	F	T	T
F	T	F	F	T	T	F	F	F
F	F	T	F	T	F	T	T	T
F	F	F	F	T	T	T	T	T

Question 3

- Construct the truth table

(P <-> Q) $\wedge$ (~(P $\wedge$ ~Q))

P	Q	P<->Q	~Q	P$\wedge$~Q	~(P$\wedge$~Q)	(P <-> Q) $\wedge$ (~(P $\wedge$ ~Q))
T	T	T	F	F	T	T
T	F	F	T	T	F	F
F	T	F	F	F	T	F
F	F	T	T	F	T	T

Types of fuzzy Propositions

- Unconditional and unqualified propositions
- Unconditional and qualified propositions
- Conditional and unqualified propositions
- Conditional and qualified propositions

Unconditional and unqualified propositions

- The canonical form of fuzzy proposition is express by the sentence

  P: $\mathcal{V}$ is F

Where,

$\mathcal{V}$: is a variable that takes values v from some universal set V.

F: is a fuzzy set on V that represents a fuzzy predicate, such as tall, low, normal and so on.

Example:

P: temperature ($\mathcal{V}$) is high (F).

Unconditional and qualified propositions

- Proposition of this type can be characterized by either the canonical form

  P: $\mathcal{V}$ is F is S

Or the canonical form

  p: Pro{$\mathcal{V}$ is F} is P

Example:

Tina is young is very true.

Conditional and unqualified propositions

- Proposition of this type can be expressed by the canonical form

  p: if $\mathcal{X}$ is A, then $\mathcal{Y}$ is B

Where $\mathcal{X}$, $\mathcal{Y}$ are variables whose values are in sets X, Y, respectively, and A, B are fuzzy sets on X, Y, respectively.

Example:

If the machine is defective THEN production is less.

Conditional and qualified propositions

- Proposition of this type can be characterized by either the canonical form

  p: if $\mathcal{X}$ is A, then $\mathcal{Y}$ is B is S

Or the canonical form

  p: Pro{$\mathcal{X}$ is A|$\mathcal{Y}$ is B } is P

Where Pro{$\mathcal{X}$ is A|$\mathcal{Y}$ is B} is a conditional probability.

Example:

If the machine is defective THEN production is less is very true.

Compositional rules of inference

Compositional rules of inference

- The inference rules of PL provide the means to perform logical proofs or deductions.
- Rules are
  - Modus ponens
  - Modus tollens
  - Chain rule
  - Substitution
  - Simplification
  - Conjunction
  - Transposition

Inference Rules cont………

Modus ponens. From P and $P \to Q$ infer Q. This is sometimes written as

$$\frac{\begin{array}{c}P\\ P \to Q\end{array}}{Q}$$

For example

given: (joe is a father)
and: (joe is a father) → (joe has a child)
conclude: (joe has a child)

Inference Rules cont………

Modus tollens

- $\alpha \rightarrow \beta$

  $\underline{\neg\beta}$

  $\neg\alpha$

Example:

Given: The machine is defective ->the production is less

And: the production is not less

Conclude: The machine is not defective

Inference Rules cont………

Chain rule. From $P \rightarrow Q$, and $Q \rightarrow R$, infer $P \rightarrow R$. Or

$$\frac{\begin{array}{c} P \rightarrow Q \\ Q \rightarrow R \end{array}}{P \rightarrow R}$$

For example,

given: (programmer likes LISP) → (programmer hates COBOL)
and: (programmer hates COBOL) → Programmer likes recursion)
conclude: (programmer likes LISP) → (programmer likes recursion)

Inference Rules cont………

Substitution. If s is a valid sentence, s' derived from s by consistent substitution of propositions in s, is also valid. For example, the sentence $P \vee \neg P$ is valid; therefore $Q \vee \neg Q$ is also valid by the substitution rule.

Inference Rules cont………

Simplification. From P & Q infer P.

Conjunction. From P and from Q, infer P & Q.

Transposition. From $P \rightarrow Q$, infer $\sim Q \rightarrow \sim P$.

Methods of decompositions

Decomposition of compound rules

- A linguistic statement expressed by a human might involve compound rule structures.
- As an example, consider a rule base for a simple home temperature control problem, which might contain the following rules.

Example

IF it is raining hard
THEN close the window.
IF the room temperature is very hot,
THEN
IF the heat is on
THEN turn the heat lower
ELSE
IF (the window is closed) AND (the air conditioner is off)
THEN (turn on the air conditioner)
AND (it is not raining hard)
THEN open the window
ELSE
IF (the window is closed) AND (the air conditioner is on)
THEN open the window; etc.

Continued…..

- By using the basic properties and operations defined for fuzzy sets, any compound rule structure may be decomposed and reduced to a number of simple canonical rules.
- These rules are based on natural language representations and models, which are themselves based on fuzzy sets and fuzzy logic.

Continued…..

- The following illustrate a number of the most common techniques for decomposition of compound linguistic rules into simple canonical forms:
  - **Multiple conjunctive antecedents**
  - **Multiple disjunctive antecedents**
  - **Conditional statements with ELSE and UNLESS**
  - **Nested IF-THEN rules**

Multiple conjunctive antecedents

This uses fuzzy intersection operation. Since it involves linguistic "AND" connective

$$\text{IF } x \text{ is } \underset{\sim}{P}^1 \text{ AND } \underset{\sim}{P}^2 \cdots \text{ AND } \underset{\sim}{P}^n \text{ THEN } y \text{ is } Q^r,$$

where

$$\underset{\sim}{P}^r = \underset{\sim}{P}^1 \text{ AND } \underset{\sim}{P}^2 \cdots \text{ OR } \underset{\sim}{P}^n.$$

The membership for this can be

$$\mu_{\underset{\sim}{P}^r}(x) = \min\left[\mu_{\underset{\sim}{P}^1}(x), \mu_{\underset{\sim}{P}^2}(x), \ldots, \mu_{\underset{\sim}{P}^n}(x)\right].$$

Hence the rule can be
IF x is $\underset{\sim}{P}^r$ THEN $\underset{\sim}{Q}^r$.

Multiple disjunctive antecedents

This uses fuzzy union operations. It involves linguistic "OR" connections
IF x is $\underset{\sim}{P}^1$ OR $\underset{\sim}{P}^2 \cdots$ OR $\underset{\sim}{P}^n$ THEN y is $\underset{\sim}{Q}^r$,
where

$$\begin{aligned}\underset{\sim}{P}^r &= \underset{\sim}{P}^1 \text{ OR } \underset{\sim}{P}^2 \cdots \text{ OR } \underset{\sim}{P}^n \\ &= \underset{\sim}{P}^1 \cup \underset{\sim}{P}^2 \cdots \cup \underset{\sim}{P}^n\,.\end{aligned}$$

The membership for this can be

$$\mu_{\underset{\sim}{P}^r}(x) = \max\left[\mu_{\underset{\sim}{P}^1}(x), \mu_{\underset{\sim}{P}^2}(x), \ldots, \mu_{\underset{\sim}{P}^n}(x)\right].$$

Hence the rule can be
IF x is $\underset{\sim}{P}^r$ THEN y is $\underset{\sim}{Q}^r$.

Conditional statements with ELSE

(a) IF $\underset{\sim}{P}^1$ THEN $\left(\underset{\sim}{Q}^1 \text{ ELSE } \underset{\sim}{Q}^2\right)$.

Considering this as one compound statement, splitting this into two canonical form rules, we get

IF $\underset{\sim}{P}^1$ THEN Q^1 OR IF NOT $\underset{\sim}{P}^1$ THEN Q^2.

(b) IF $\underset{\sim}{P}^1$ THEN $\left(Q^1 \text{ ELSE } \underset{\sim}{\underset{\sim}{P}^2 \text{ THEN }} \left(\underset{\sim}{Q}^2\right)\right)$.

The decomposition for this can be of the form

$$\text{IF } \underset{\sim}{P}^1 \text{ THEN } Q^1 \text{ OR}$$

$$\text{IF NOT } \underset{\sim}{P}^1 \text{ AND } \underset{\sim}{P}^2 \text{ THEN } Q^2.$$

Nested IF-THEN rules

$$\text{IF } \underset{\sim}{P}^1 \text{ THEN } \left(\text{IF } \underset{\sim}{P}^2 \text{ THEN } (\underset{\sim}{Q}^2)\right).$$

This can be decomposed into

$$\text{IF } \underset{\sim}{P}^1 \text{ AND } \underset{\sim}{P}^2 \text{ THEN } Q^1.$$

Thus the compound rules are decomposed into single canonical rules. Then this rules may be reduced to a series of relations.

Defuzzification

Defuzzification

- In many situations, for a system whose output is fuzzy, it is easier to take a crisp decision if the output is represented as a single scalar quantity.
- The conversion of a fuzzy set to single crisp value is called defuzzification.
- It is a reverse process of fuzzification.

Defuzzification techniques

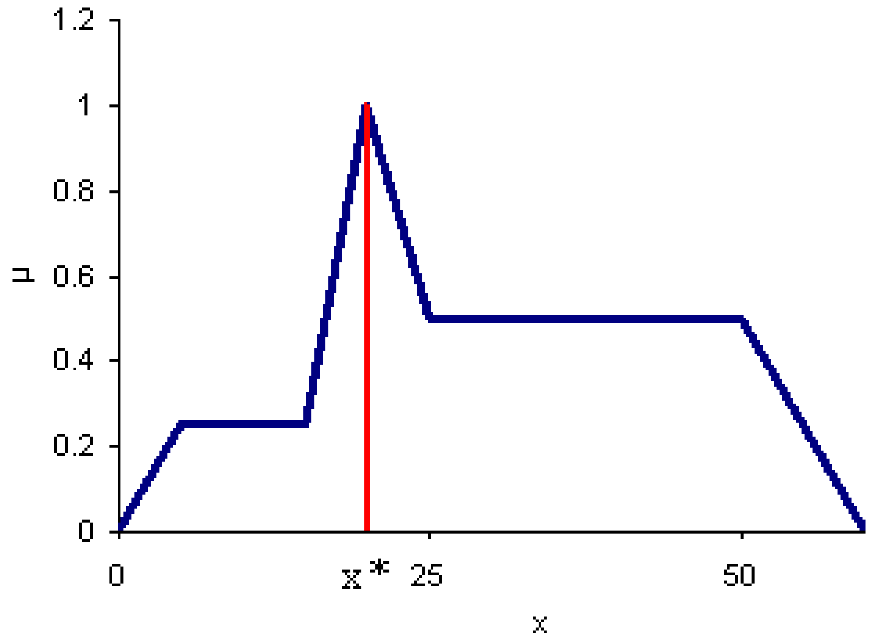
1. Maximum Defuzzification Technique
2. Centroid Defuzzification Technique
3. Weighted Average Defuzzification Technique
4. Mean-Max Membership
5. Centre of sum
6. Centre of singleton Method

Maximum Defuzzification Technique

- This method gives the output with the highest membership function.
- This defuzzification technique is very fast but is only accurate for peaked output.
- This technique is given by algebraic expression as

$$\mu_{A}(x^{*}) \geq \mu_{A}(x) \quad \text{for all } x \in X$$

Maximum Defuzzification Technique



where x* is the defuzzified value.

Centroid Defuzzification Technique

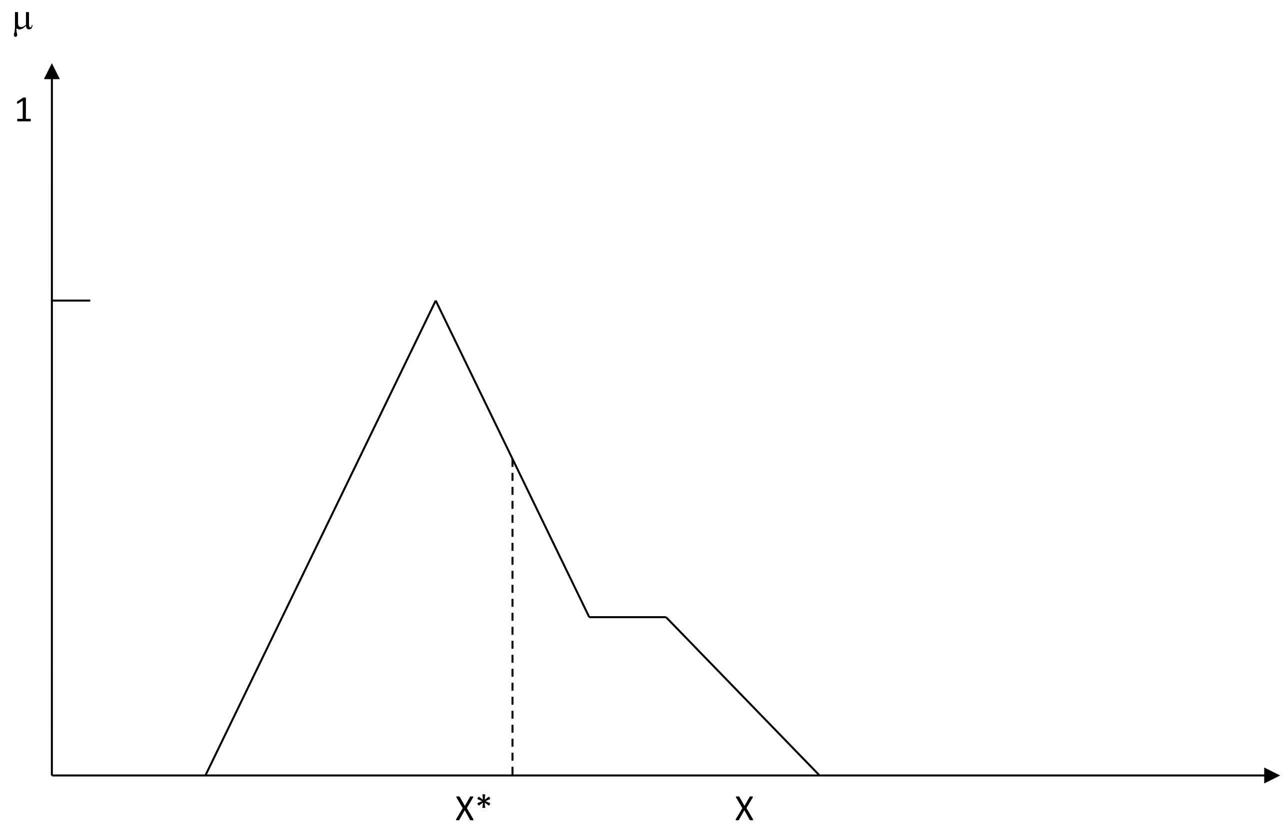
- This method is also known as Center Of Gravity (COG) or Center Of Area (COA) defuzzification.
- This technique was developed by Sugeno in 1985.
- This is the most commonly used technique and is very accurate.
- The centroid defuzzification technique can be expressed as

Centroid Defuzzification Technique

$$x^* = \frac{\int \mu_i(x)\, x\, dx}{\int \mu_i(x)\, dx}$$

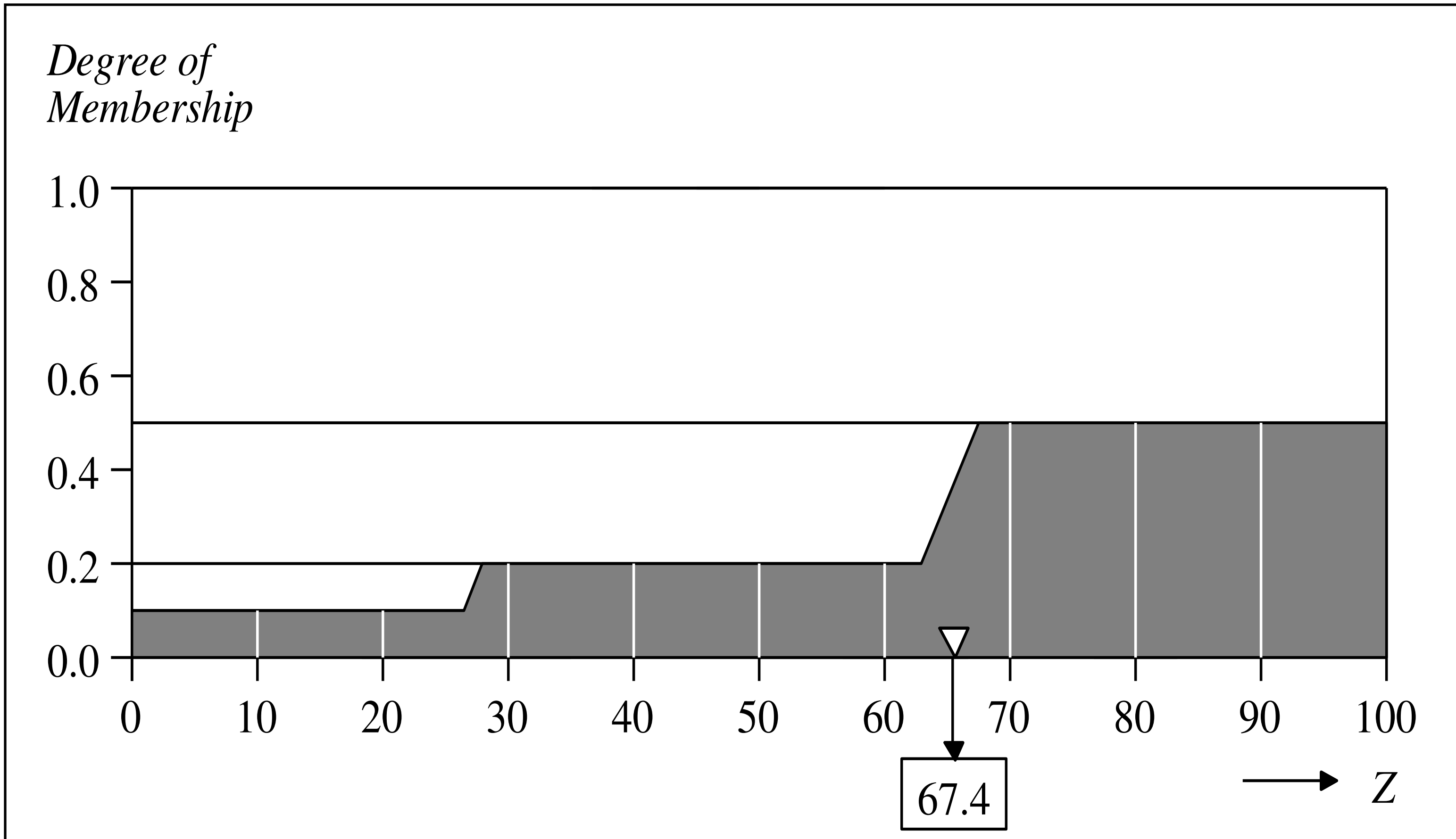
- where x^* is the defuzzified output, $\mu_i(x)$ is the aggregated membership function and x is the output variable.
- The only disadvantage of this method is that it is computationally difficult for complex membership functions.

Centroid Defuzzification Technique



Centroid Defuzzification Technique

$$COG = \frac{(0+10+20)\times 0.1+(30+40+50+60)\times 0.2+(70+80+90+100)\times 0.5}{0.1+0.1+0.1+0.2+0.2+0.2+0.2+0.5+0.5+0.5+0.5} = 67.4$$



Weighted Average Defuzzification

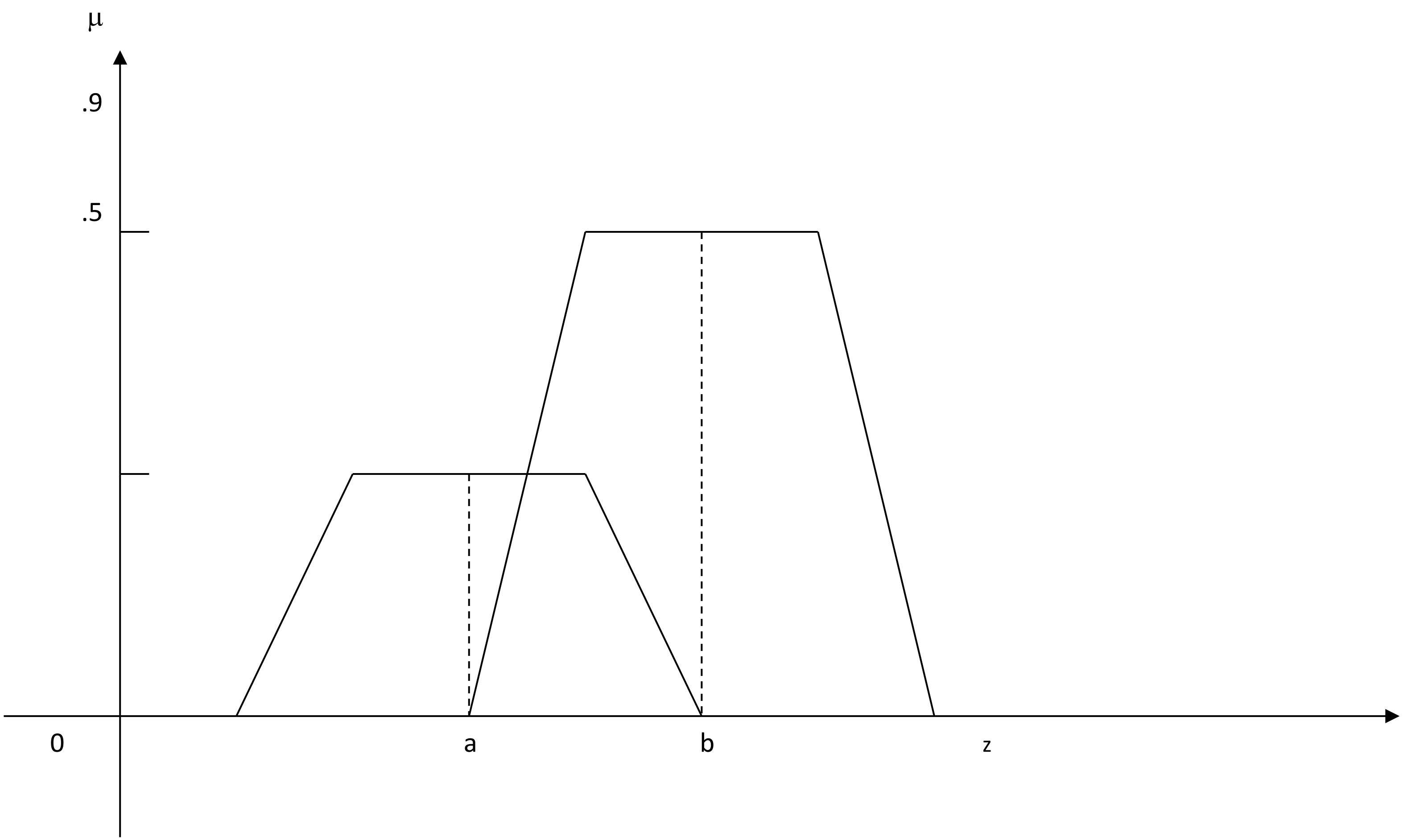
- In this method the output is obtained by the weighted average of the each output of the set of rules stored in the knowledge base of the system.
- The weighted average defuzzification technique can be expressed as

Weighted Average Defuzzification

$$x^* = \frac{\sum_{i=1}^{n} m^i w_i}{\sum_{i=1}^{n} m^i}$$

- where x^* is the defuzzified output, m^i is the membership of the output of each rule, and w_i is the weight associated with each rule. This method is computationally faster and easier and gives fairly accurate result

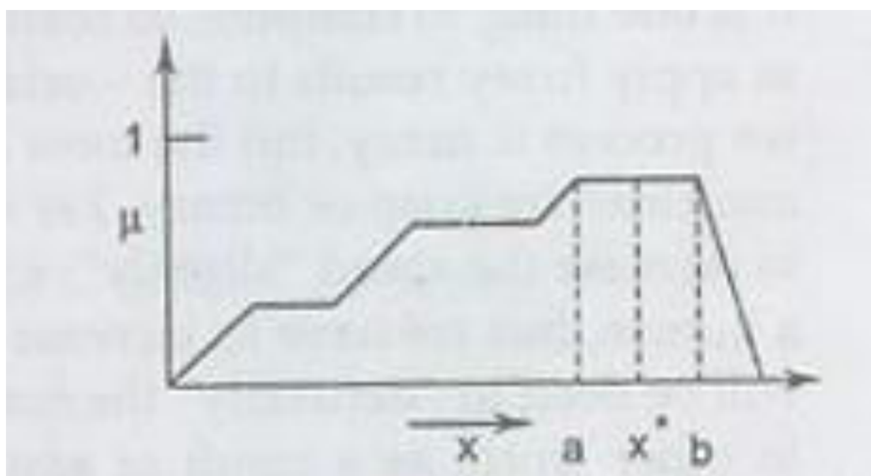
Weighted Average Defuzzification



Mean-Max Membership

- This method is closely related to the first method, except that the location of maximum membership can be non-unique.
- It is given by expression:

$$x^* = \frac{a+b}{2}$$



Centre of sum

- This process involves the algebraic sum of individual output fuzzy sets, instead of their union.
- The difuzzified value x* is given by:

$$x^* = \frac{\int_x x \sum_{k=1}^{n} \mu_{A_K}(x)dx}{\int_x \sum_{k=1}^{n} \mu_{A_K}(x)dx}$$

Centre of sum

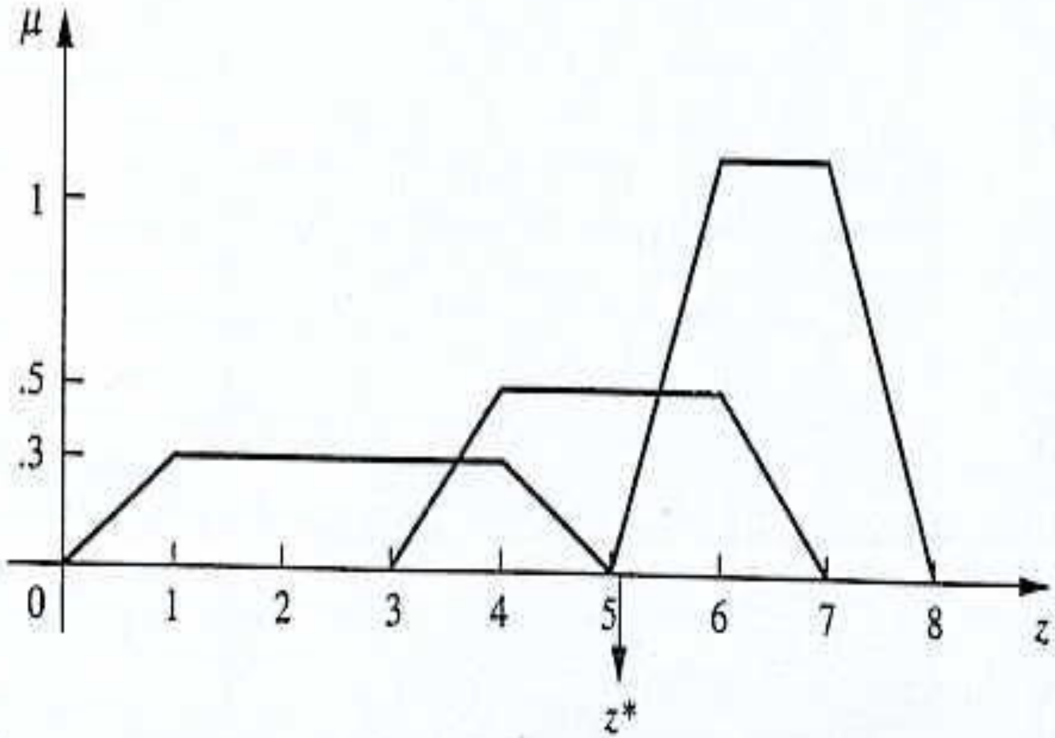


FIGURE 5.24
Center of sums result for Example 5.5.

Centre of singleton Method

- Defuzzification can be strongly simplified if the membership functions $\mu_A(x)$ of the conclusions are singly defuzzified for each rule such that each function is reduced to a singleton that has the position μ'_A *of the individual membership function's centre of gravity.*
- *The centre of singleton is calculated by using the degree of relevance as follows:*

$$\mu'_A = \frac{\sum_A \mu'_A X_A}{\sum_A X_A}$$

Example

The data for 3 fuzzy sets is shown in figure. Difuzzify the data using weighted average method and mean-max membership method.

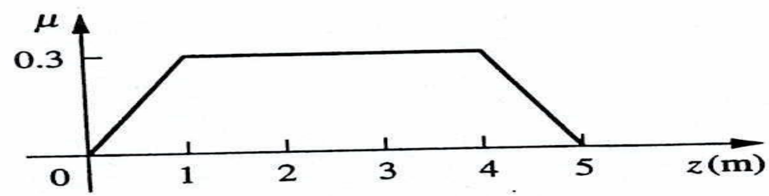


FIGURE 5.9
Fuzzy set $\underset{\sim}{B}_1$: Public right-of-way width (z) for survey 1.

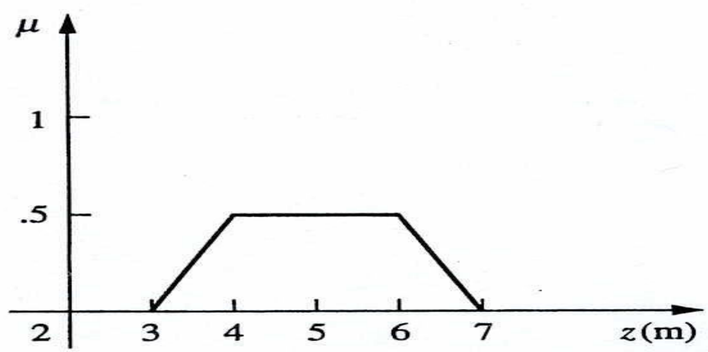


FIGURE 5.10
Fuzzy set $\underset{\sim}{B}_2$: Public right-of-way width (z) for survey 2.

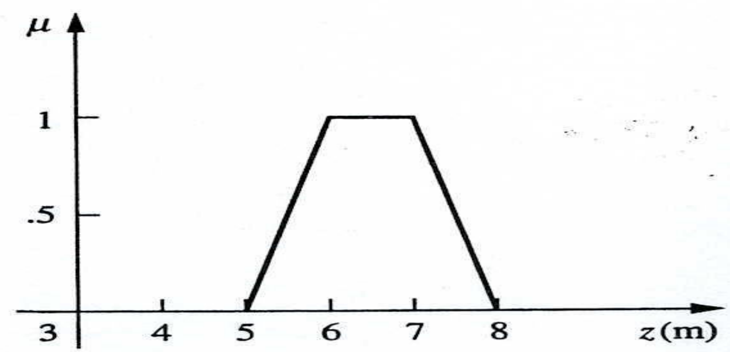


FIGURE 5.11
Fuzzy set $\underset{\sim}{B}_3$: Public right-of-way width (z) for survey 3.

Solution

Weighted-Average Method:

$$z^* = \frac{(.3\times 2.5)+(.5\times 5)+(1\times 6.5)}{.3+.5+1} = 5.41 meters$$

Mean-Max Method: $(6+7)/2 = 6.5 meters$

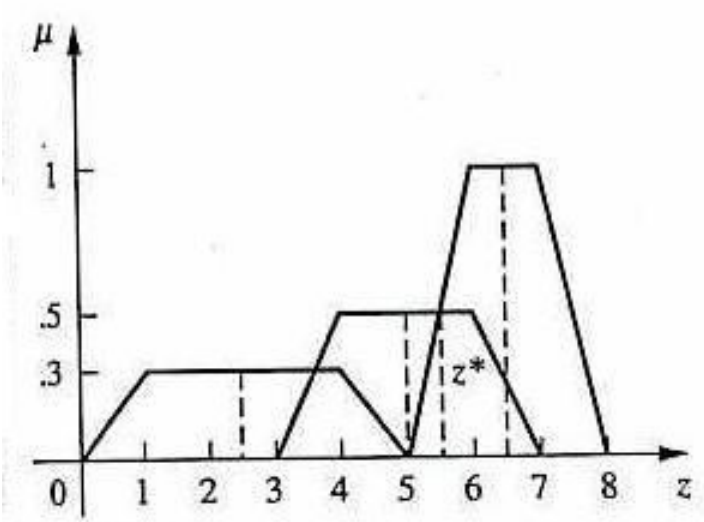


FIGURE 5.13
The weighted average method for finding z^*.

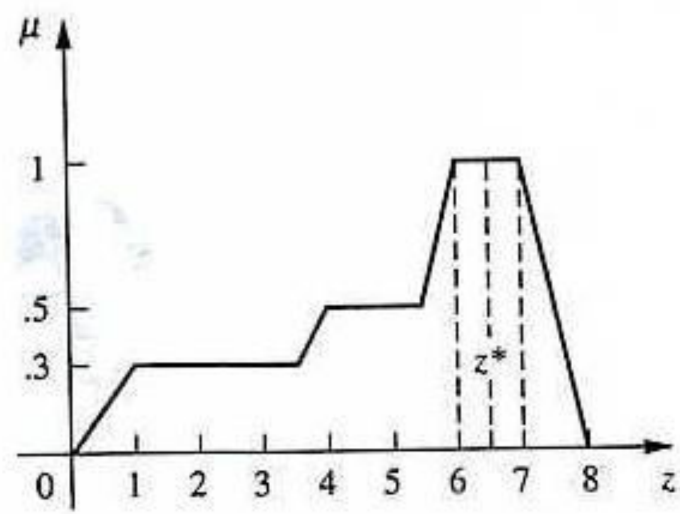


FIGURE 5.14
The mean-max membership method for finding z^*.

Centre of sum

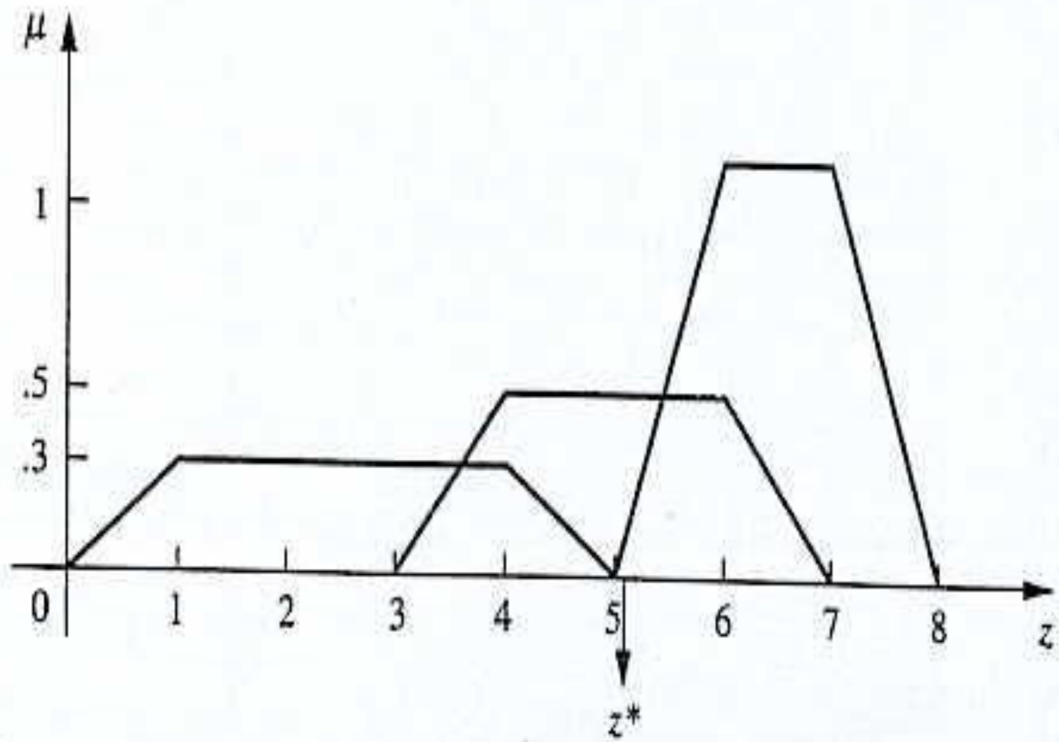


FIGURE 5.24
Center of sums result for Example 5.5.

Fuzzy inference system

- The advantage of a fuzzy set approach is that it can usefully describe imprecise, incomplete or vague information.
- However, being able to describe such information is of little practical use unless we can infer with it.
- The accepted method of the application of fuzzy sets is analogous to, but different from, the way a conventional knowledge base system is organized.
- Assuming that there is a particular problem that cannot be tackled by conventional methods such as by developing a mathematical model, after some process, the 'base' fuzzy sets that describe the problem are determined.
- After that the IF-THEN rules are thus determined.
- These rules then have to be combined in some way referred to as rule composition.
- Finally conclusions have to be drawn.

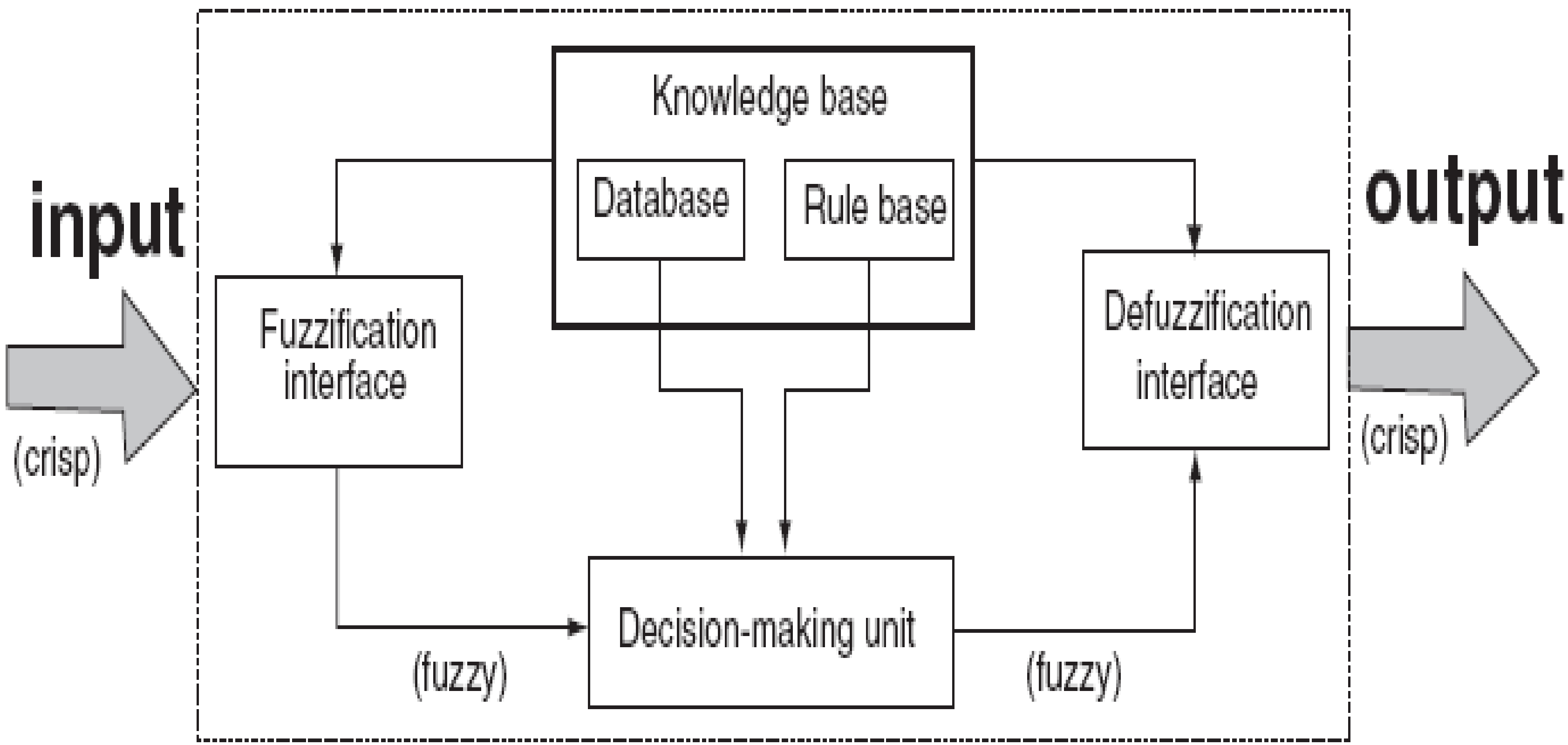
Fuzzy inference system cont……….

- Thus it is a process of formulating the mapping from a given input to an output using fuzzy logic.
- Fuzzy inference system have been successfully applied in fields such as automatic control, data classification, decision analysis, expert systems.
- Because of its multidisciplinary nature, fuzzy inference systems are associated with a number of names, such as fuzzy-rule-based systems, fuzzy expert systems, fuzzy modeling, fuzzy associative memory, fuzzy logic controllers, and simply fuzzy systems.

Fuzzy inference system cont……….

- Fuzzy inference system consists of five functional blocks
  - A **rule** base containing a number of fuzzy if-then rules;
  - A **database** which defines the membership functions of the fuzzy sets used in the fuzzy rules,
  - A **decision making** unit which perform inference operations on the rules
  - A **fuzzification** interface which transforms the crisp inputs into degree of match with linguistics values
  - a **defuzzification** interface which transform the fuzzy results of the interface into a crisp output.

Fuzzy inference system



Different fuzzy inference techniques

- Mamdani Fuzzy Inference
- Sugeno Fuzzy Inference
- Tsukamoto Fuzzy Inference

Mamdani Fuzzy Inference

- 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.

- The Mamdani-style fuzzy inference process is performed in four steps:
  1. Fuzzification of the input variables
  2. Rule evaluation (inference)
  3. Aggregation of the rule outputs (composition)
  4. Defuzzification.

Mamdani Fuzzy Inference

We examine a simple two-input one-output problem that includes three rules:

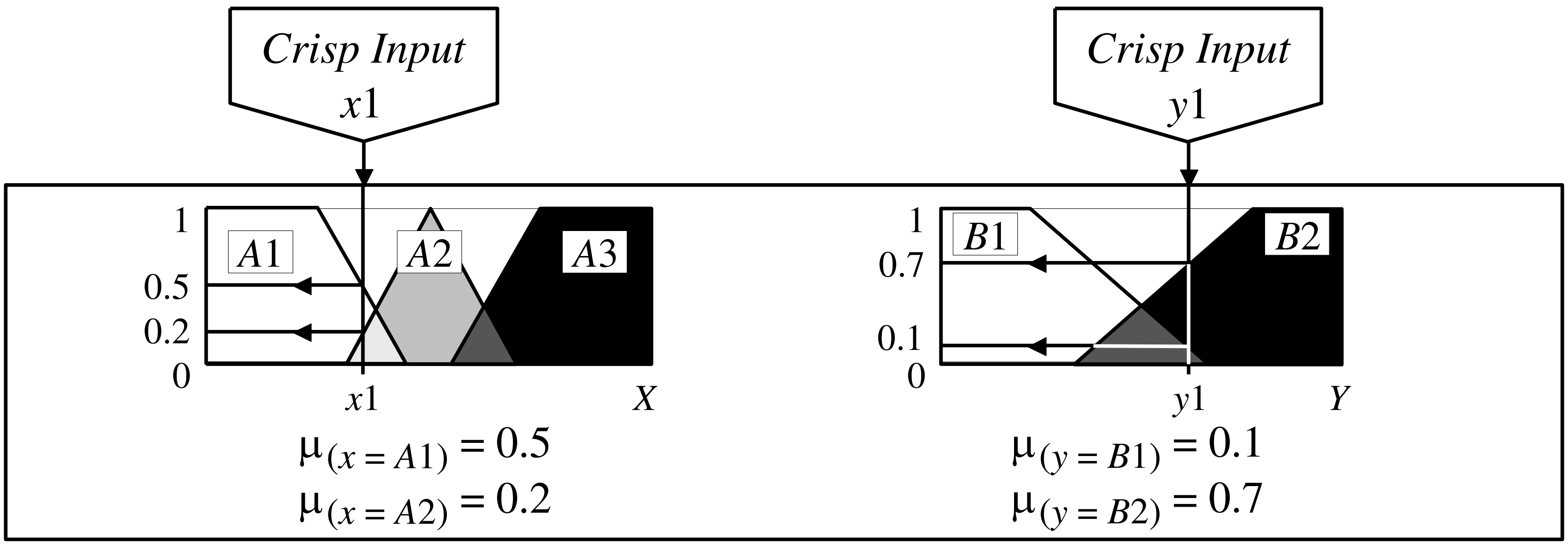
<u>Rule: 1</u>	IF x is A3	OR	y is B1	THEN	z is C1
<u>Rule: 2</u>	IF x is A2	AND	y is B2	THEN	z is C2
<u>Rule: 3</u>	IF x is A1			THEN	z is C3

Real-life example for these kinds of rules:

<u>Rule: 1</u>	IF project_funding is adequate	OR	project_staffing is small	THEN	risk is low
<u>Rule: 2</u>	IF project_funding is marginal	AND	project_staffing is large	THEN	risk is normal
<u>Rule: 3</u>	IF project_funding is inadequate			THEN	risk is high

Step 1: Fuzzification

- The first step is to take the crisp inputs, x1 and y1 (*project funding* and *project staffing*), and determine the degree to which these inputs belong to each of the appropriate fuzzy sets.



Step 2: Rule Evaluation

- The second step is to take the fuzzified inputs, $\mu_{(x=A1)} = 0.5$, $\mu_{(x=A2)} = 0.2$, $\mu_{(y=B1)} = 0.1$ and $\mu_{(y=B2)} = 0.7$, 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.

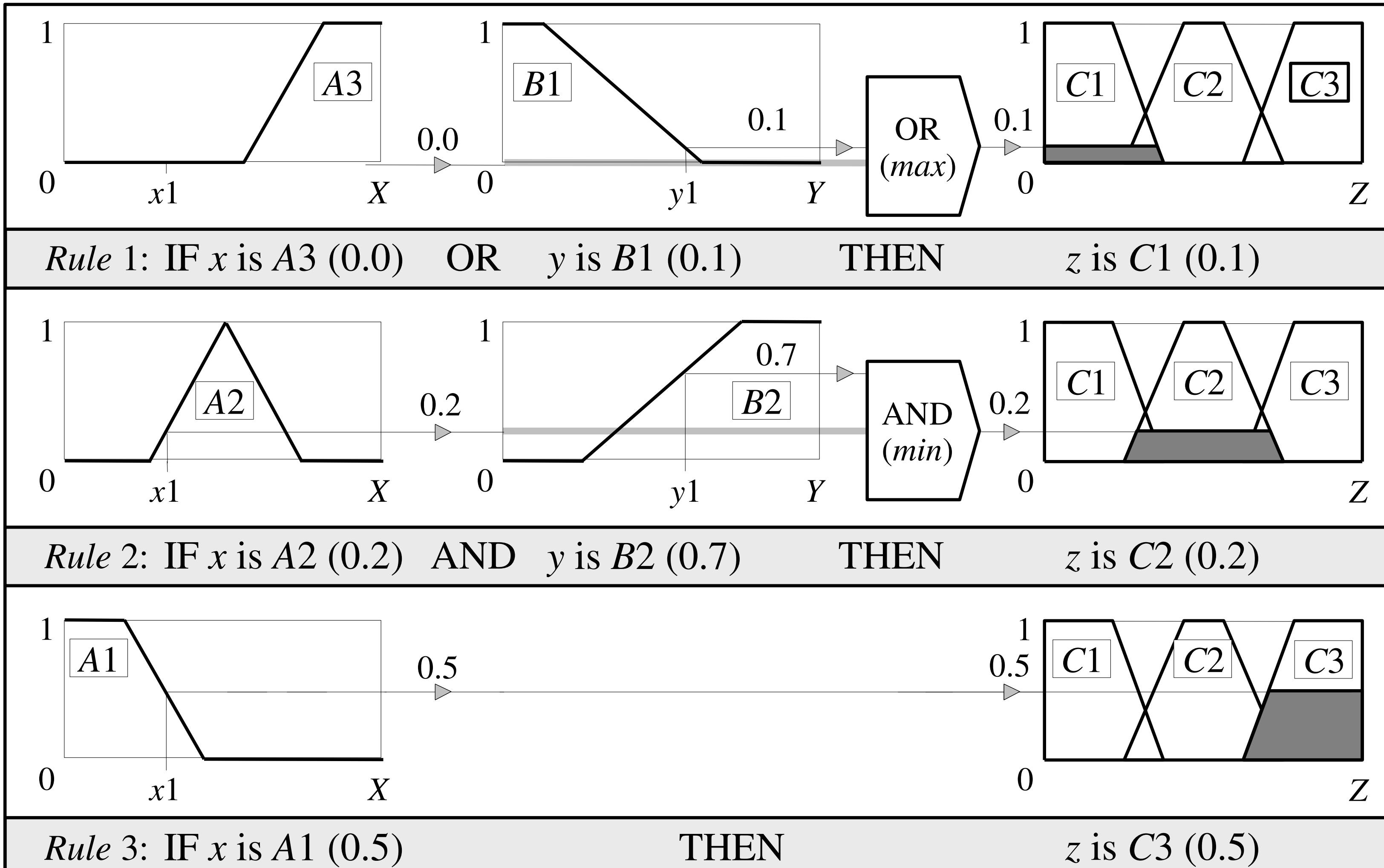
RECALL: To evaluate the disjunction of the rule antecedents, we use the **OR** fuzzy operation. Typically, fuzzy expert systems make use of the classical fuzzy operation union:

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

Similarly, in order to evaluate the conjunction of the rule antecedents, we apply the **AND** fuzzy operation intersection:

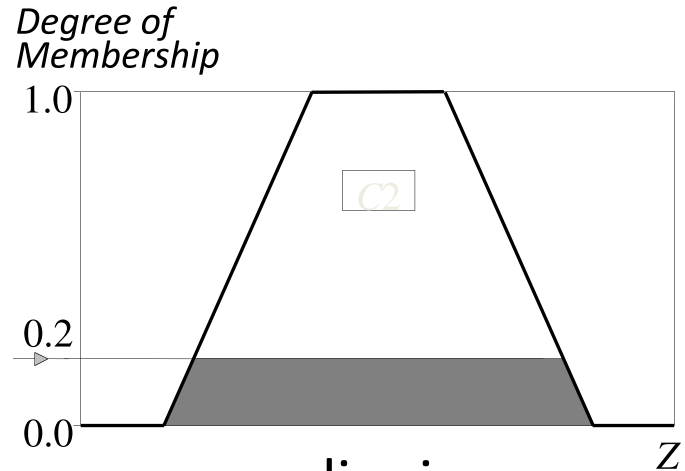
$\mu_{A \cap B}(x) = \min\,[\mu_A(x), \mu_B(x)]$

Step 2: Rule Evaluation

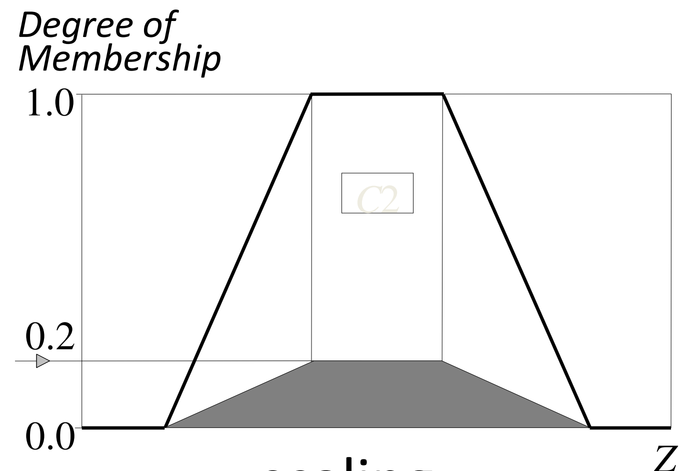


Step 2: Rule Evaluation

- Now the result of the antecedent evaluation can be applied to the membership function of the consequent.
- The most common method is to cut the consequent membership function at the level of the antecedent truth. This method is called **clipping** (alpha-cut).
  - Since the top of the membership function is sliced, the clipped fuzzy set loses some information.
  - However, clipping is still often preferred because it involves less complex and faster mathematics, and generates an aggregated output surface that is easier to defuzzify.
- While clipping is a frequently used method, **scaling** offers a better approach for preserving the original shape of the fuzzy set.
  - The original membership function of the rule consequent is adjusted by multiplying all its membership degrees by the truth value of the rule antecedent.
  - This method, which generally loses less information, can be very useful in fuzzy expert systems.



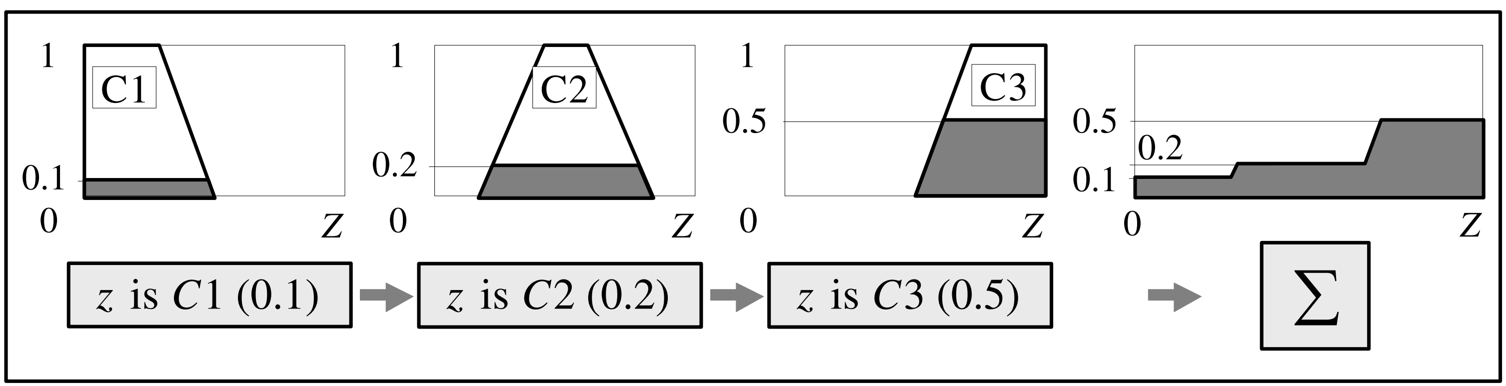
clipping



scaling

Step 3: Aggregation of the Rule Outputs

- 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.
- The input of the aggregation process is the list of clipped or scaled consequent membership functions, and the output is one fuzzy set for each output variable.



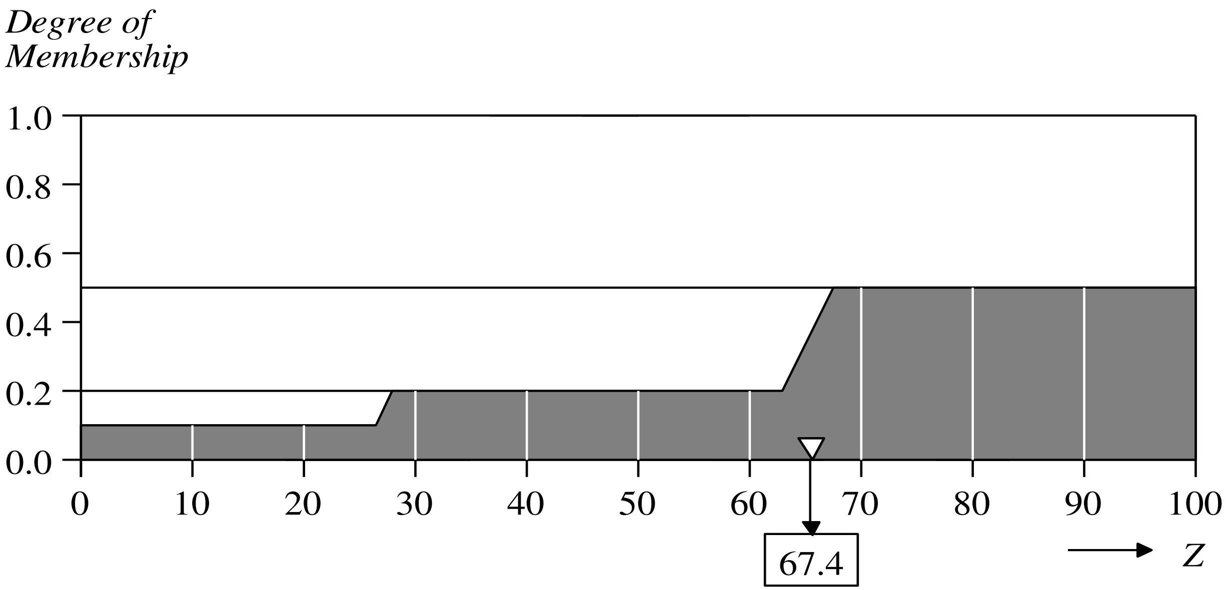
Step 4: Defuzzification

- 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.
- There are several defuzzification methods, but probably the most popular one is the **centroid technique**. It finds the point where a vertical line would slice the aggregate set into two equal masses. Mathematically this **centre of gravity** (**COG**) can be expressed as:

$$COG = \frac{\int_a^b \mu_A(x)\, x\, dx}{\int_a^b \mu_A(x)\, dx}$$

Step 4: Defuzzification

- Centroid defuzzification method finds a point representing the centre of gravity of the aggregated fuzzy set *A*, on the interval [*a, b*].
- A reasonable estimate can be obtained by calculating it over a sample of points.



$$COG = \frac{(0+10+20)\times 0.1 + (30+40+50+60)\times 0.2 + (70+80+90+100)\times 0.5}{0.1+0.1+0.1+0.2+0.2+0.2+0.2+0.5+0.5+0.5+0.5} = 67.4$$

Sugeno Fuzzy Inference

- Mamdani-style inference, as we have just seen, requires us to find 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 consequent.

- 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.

Sugeno Fuzzy Inference

- Sugeno-style fuzzy inference is very similar to the Mamdani method.
- Sugeno changed only a rule consequent: instead of a fuzzy set, he used a mathematical function of the input variable.
- The format of the **Sugeno-style fuzzy rule** is

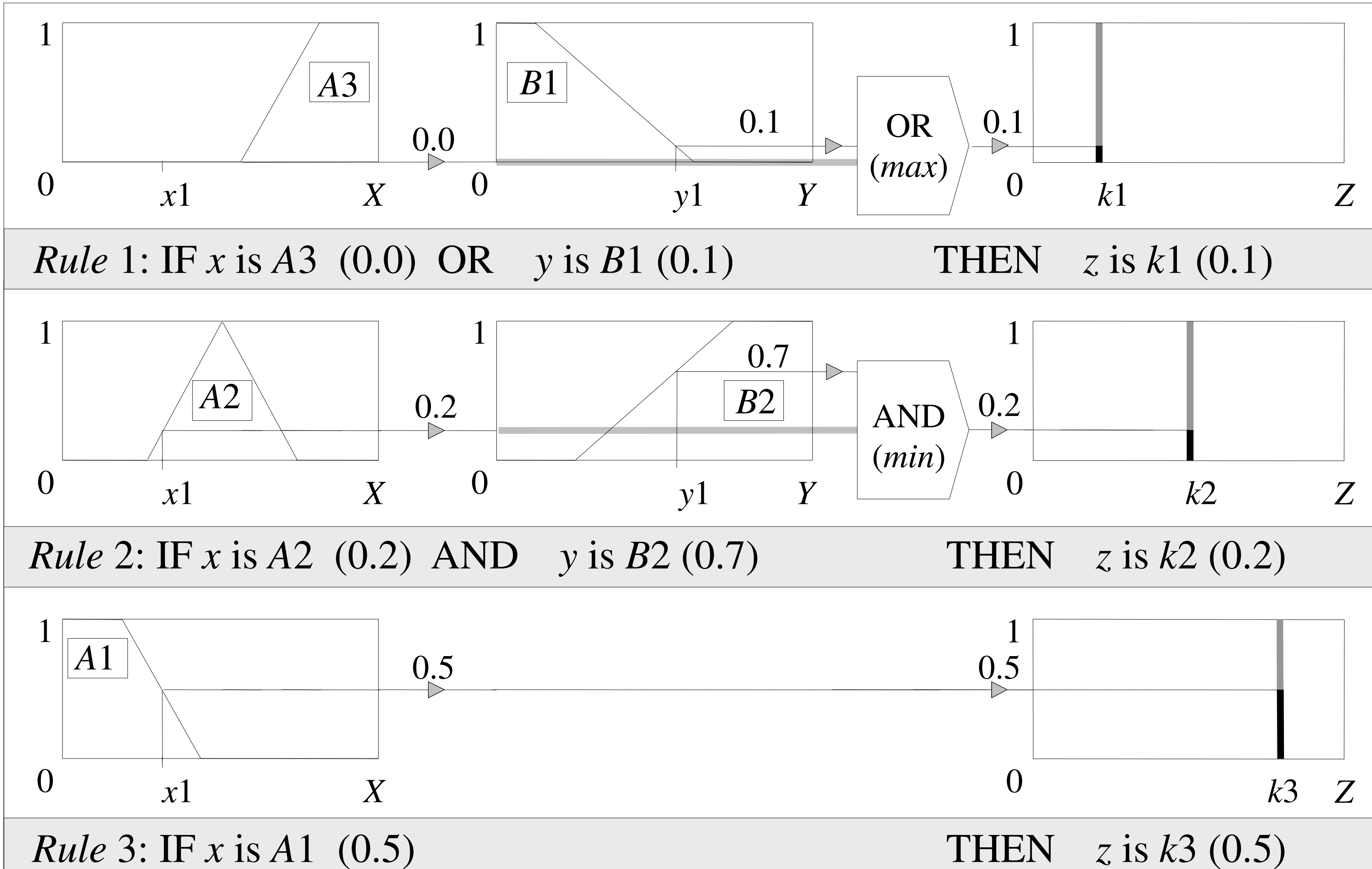
  IF ***x* is *A*** **AND** ***y* is *B*** **THEN** ***z* is *f*(*x*, *y*)**

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

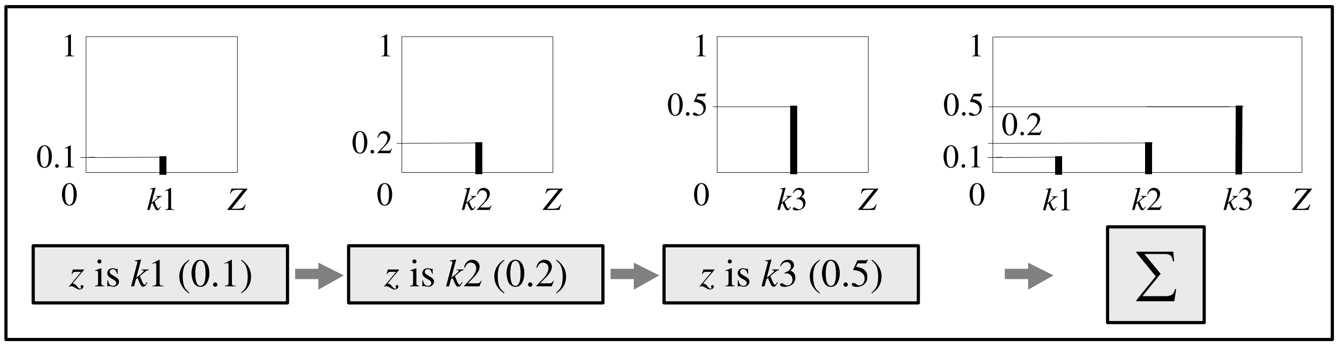
- The most commonly used **zero-order Sugeno fuzzy model** applies fuzzy rules in the following form:

  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 and all consequent **membership functions** are represented by **singleton spikes.**

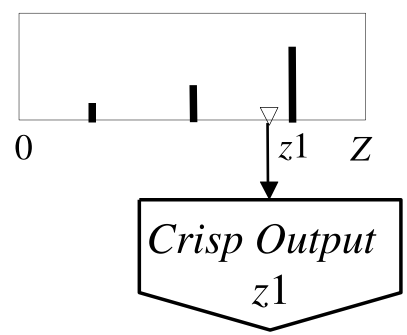
Sugeno Rule Evaluation



Sugeno Aggregation and Defuzzification



COG becomes Weighted Average (WA)



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

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.

Advantages of the Sugeno method

- It is computationally efficient.
- It works with linear techniques.
- It works well with optimization and adaptive techniques.
- It has guaranteed continuity of the output surface.
- It is well suited to mathematical analysis.

Advantages of the Mamdani method

- It is intuitive.
- It has widespread acceptance.
- It is well suited to human input.