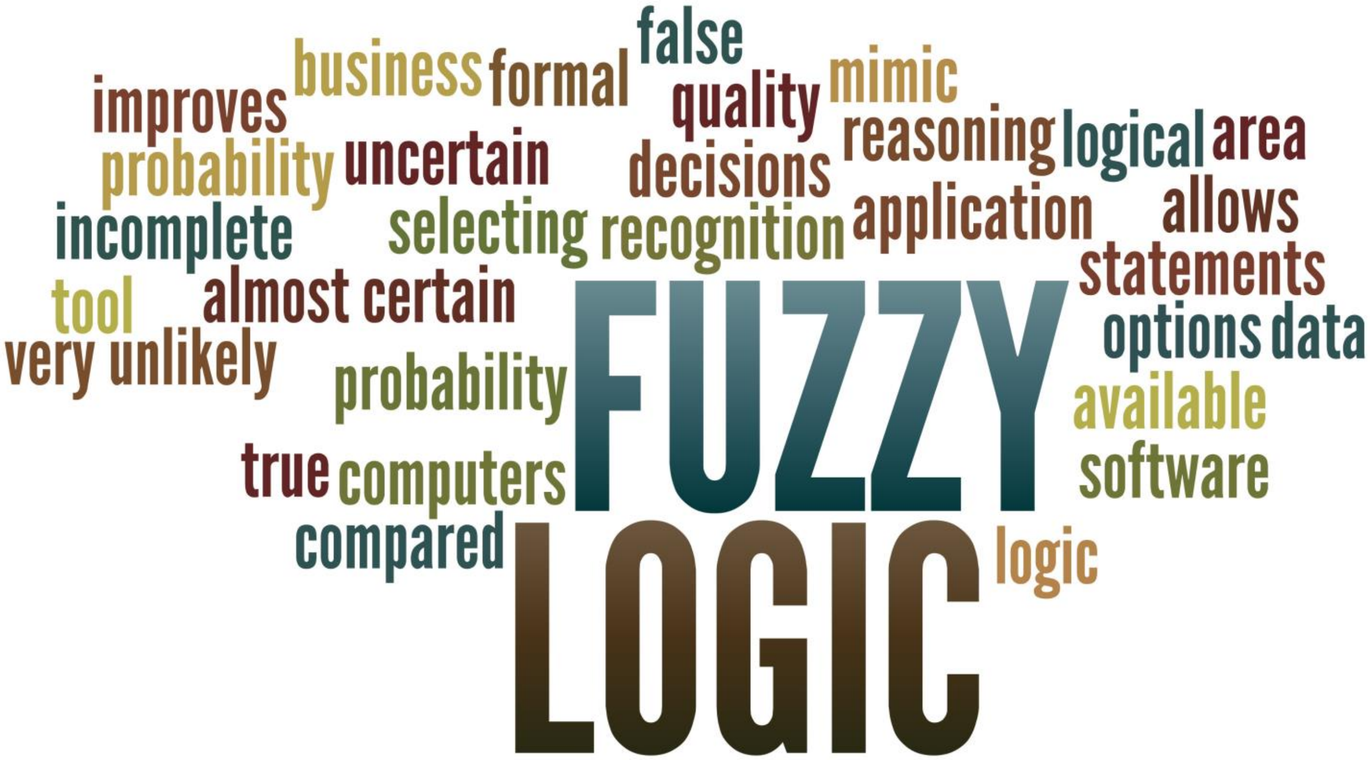
# 20IS603 Architecture of Intelligent Systems

Fuzzy Sets




Lecture #8

# Fuzzy Sets

- If *F* is a fuzzy set, then the membership function $\mu_F(x)$ measures the degree to which an absolute value $x$ belongs to *F*.

$$\mu_F(x) : X \rightarrow [0,1]$$

$$\mu_F(x) = \begin{cases} 1, & \text{if } x \text{ is totaly in } A \\ 0, & \text{if } x \text{ is not in } A \\ \in (0,1), & \text{if } x \text{ is partialy in } A \end{cases}$$

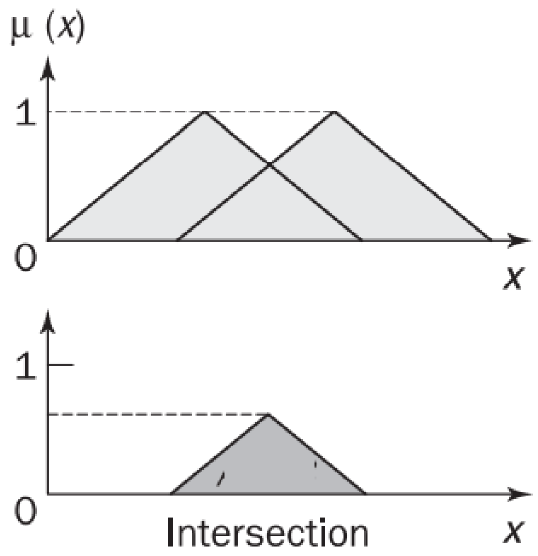
- $0 < \mu_F(x) < 1$ allows a continuum of possible choices - represents the degree of membership, also called membership value.

# Fuzzy Operations

- Basic operations for fuzzy reasoning and fuzzy control.
- Suppose X and Y are two fuzzy sets within the universe of discourse

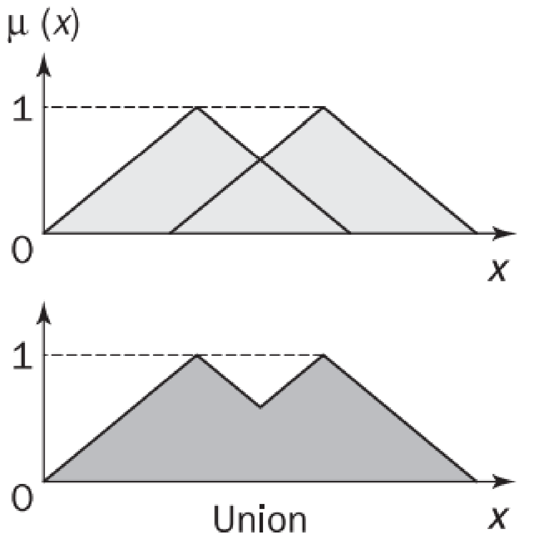
- Conjunction: The intersection of two fuzzy sets X and Y, written as X ∩ Y (X *and* Y), then MF of fuzzy set related to those of X and Y as

$$\mu_{X \text{ and } Y} = \min[\mu_X , \mu_Y]$$

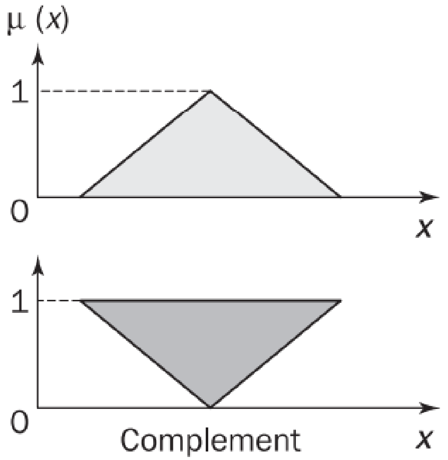



- Disjunction: The union of two fuzzy sets X and Y, X U Y (X *or* Y), then MF is

$$\mu_{X \text{ or } Y} = \max[\mu_X , \mu_Y]$$



# Fuzzy Operations



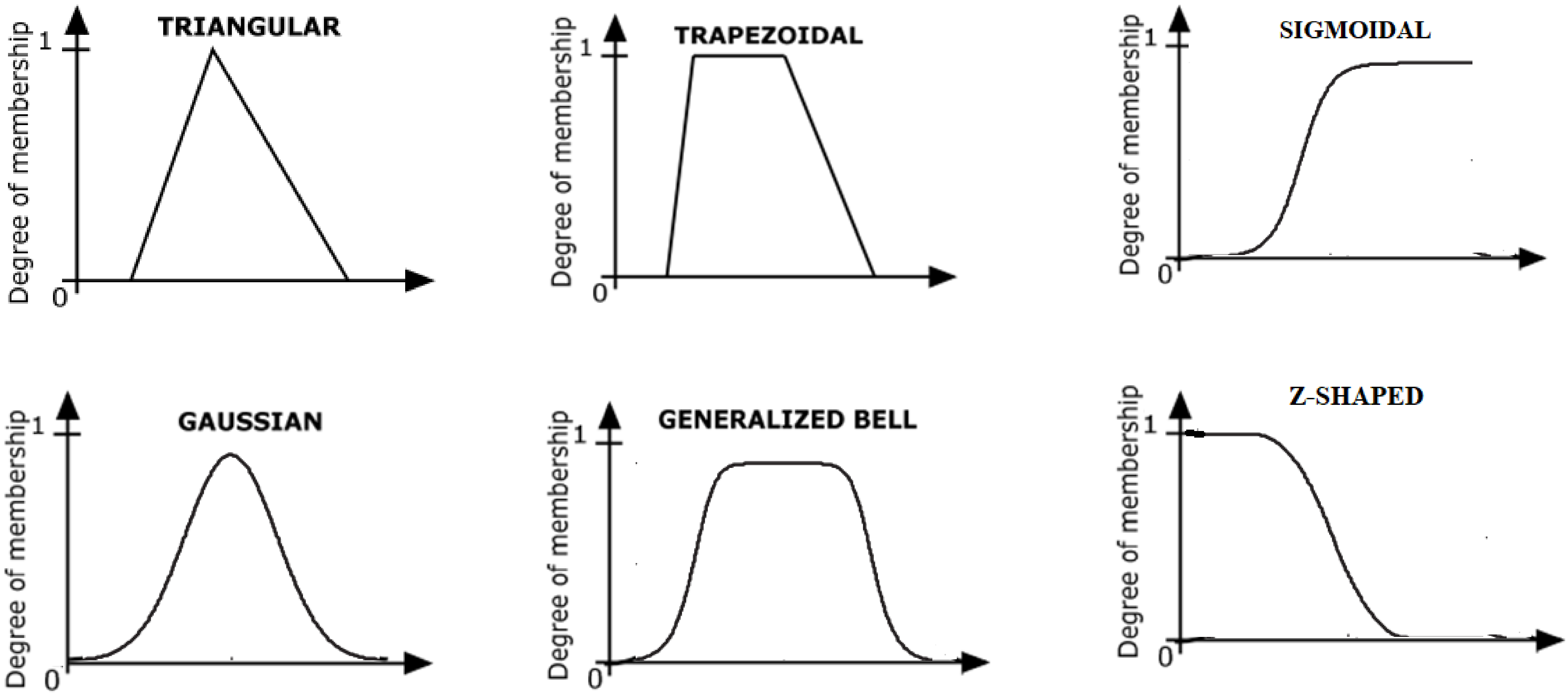

- Negation: The complement of fuzzy set X , then MF is

$$\mu_{\text{not } x} = 1 - \mu_x$$

- If several rules affect the same fuzzy set of the same variable, they are equivalent to a single rule whose conditions are joined by the disjunction *or* - the memberships are clearly dependent on each other - new operator *dor* for "*dependent or*."

$$\mu_{x \text{ dor } Y} = \min[1, \mu_x + \mu_Y]$$

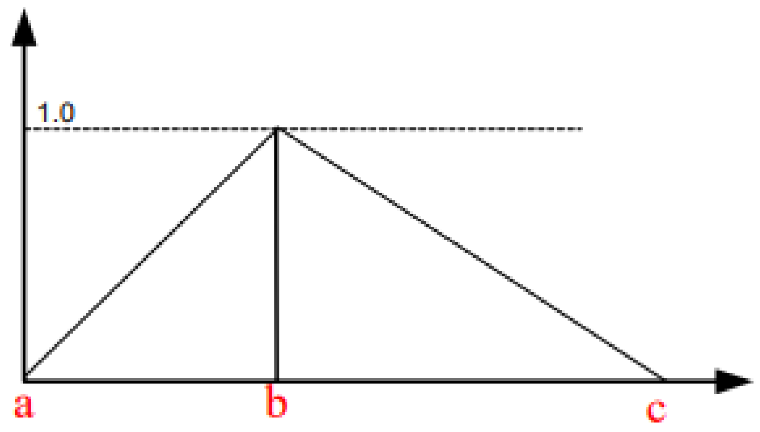
# Fuzzy Membership Functions



# Fuzzy Membership Functions

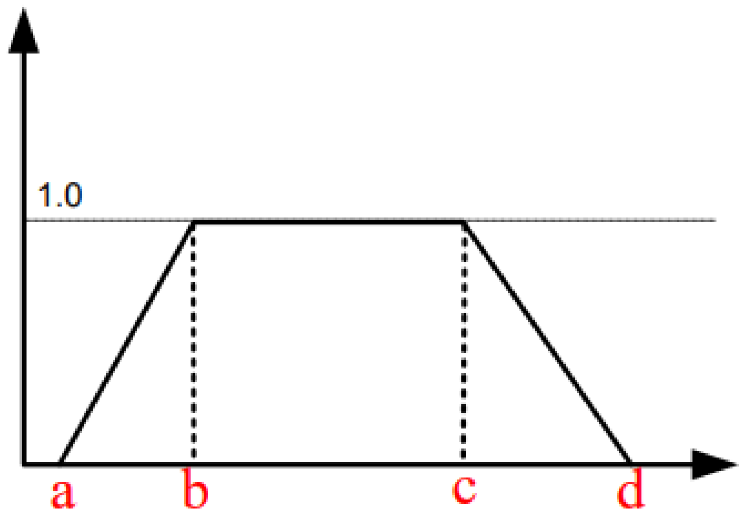
- Triangular Membership Function

$$\text{trimf}(x{:}\, a,\, b,\, c) = \begin{cases} 0, & x \le a \\ \dfrac{x-a}{b-a}, & a \le x \le b \\ \dfrac{c-x}{c-b}, & b \le x \le c \\ 0, & c \le x \end{cases}$$




- Trapezoidal Membership Function

$$\text{trapmf}(x{:}\, a,\, b,\, c) = \begin{cases} 0, & x \le a \\ \dfrac{x-a}{b-a}, & a \le x \le b \\ 1, & b \le x \le c \\ \dfrac{d-x}{d-c}, & c \le x \le d \\ 0, & d \le x \end{cases}$$

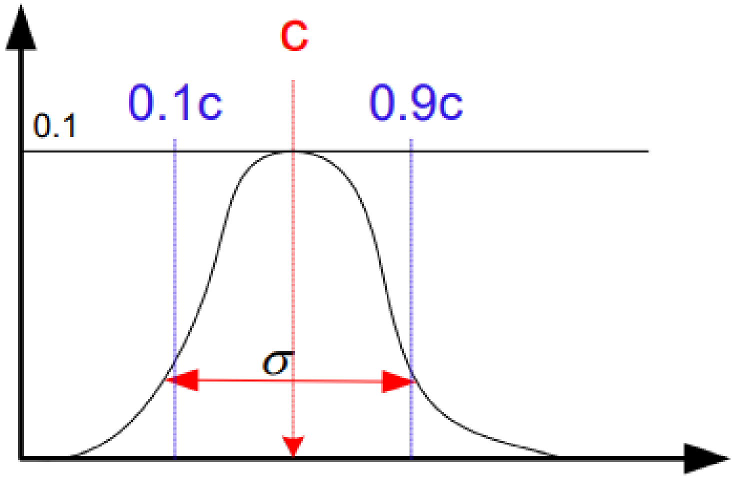


# Fuzzy Membership Functions

- Gaussian Membership Function

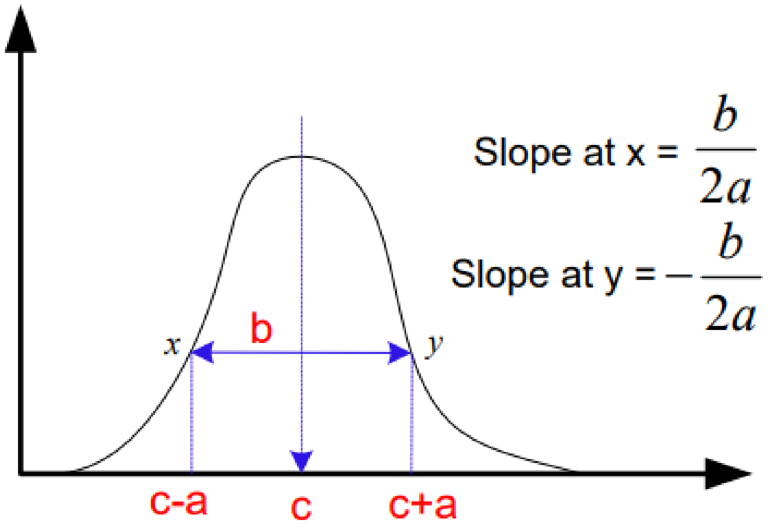
$$\text{gaussmf}(x\colon \sigma, c) = \exp\left[\frac{-(x-c)^2}{2\sigma^2}\right]$$

c determines the center of the MF
σ represents the width




- Generalized Bell Membership Function

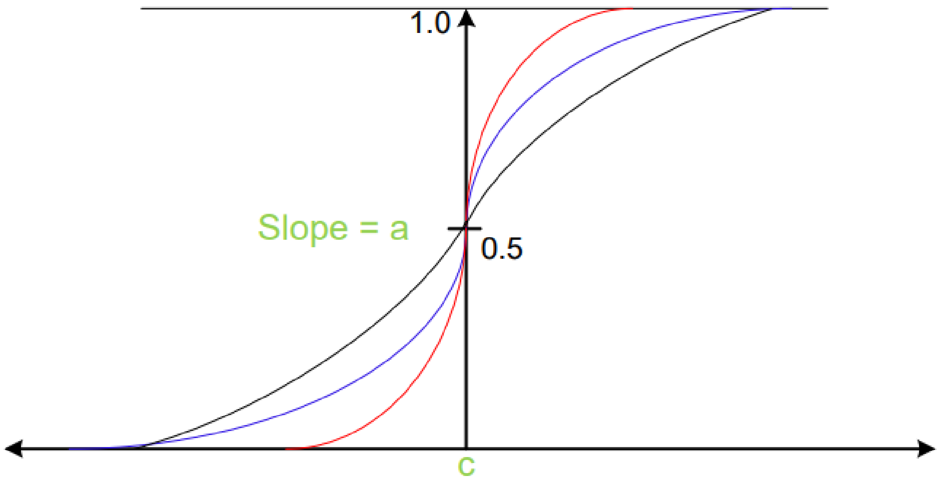
$$\text{gbellmf}(x\colon a, b, c) = \frac{1}{1+\left|\frac{x-c}{a}\right|^{2b}}$$



# Fuzzy Membership Functions

- Sigmoidal Membership Function

$$\mathrm{sigmf}(x{:}\,a,c) = \frac{1}{1+e^{-a(x-c)}}$$




- Z-Shaped Membership Function

$$\mathrm{zmf}(x{:}\,a,\,b) = \begin{cases} 1, & x \le a \\ 1-2\left(\frac{x-a}{b-a}\right)^2, & a \le x \le \frac{a+b}{2} \\ 2\left(b-\frac{x}{b-a}\right)^2, & \frac{a+b}{2} \le x \le b \\ 0, & b \le x \end{cases}$$

# Defuzzification

- Interpret the membership values in terms of a numerical value
- Important only when the fuzzy variable is a control action where a specific setting is required
- Stage 1: Scaling the Membership Functions
  - To adjust the fuzzy sets in accordance with the calculated possibilities
  - Larsen's product operation rule - the membership functions are multiplied by their respective possibility values

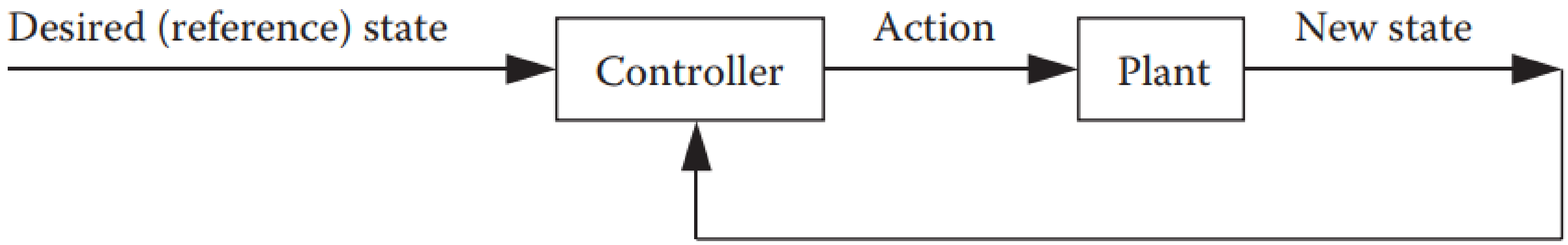
$$\mu_i A_i$$

- Stage 2: Finding the Centroid
  - Also called the center of gravity, center of mass, or center of area method (COG)
  - Determining the centroid of the fuzzy output function is part of an overall process known as Mamdani-style fuzzy inference
  - If there are N membership functions with centroids $c_i$ and areas $a_i$ , then the combined centroid C, that is, the defuzzified value is,

$$C = \frac{\sum_{i=1}^{N} a_i c_i}{\sum_{i=1}^{N} a_i}$$

# Fuzzy Control Systems

- Control decisions can be thought of as a transformation from state variables to action variables.




- *State variables* - the current state of the physical plant and the desired state
- *Action variables* - those that can be directly altered by the controller
- Fuzzy logic enables a small number of rules to produce smooth changes in the action variables as the state variables change.
- Well suited to control decisions where the control actions need to be scaled as input measurements change.

# Fuzzy Rules

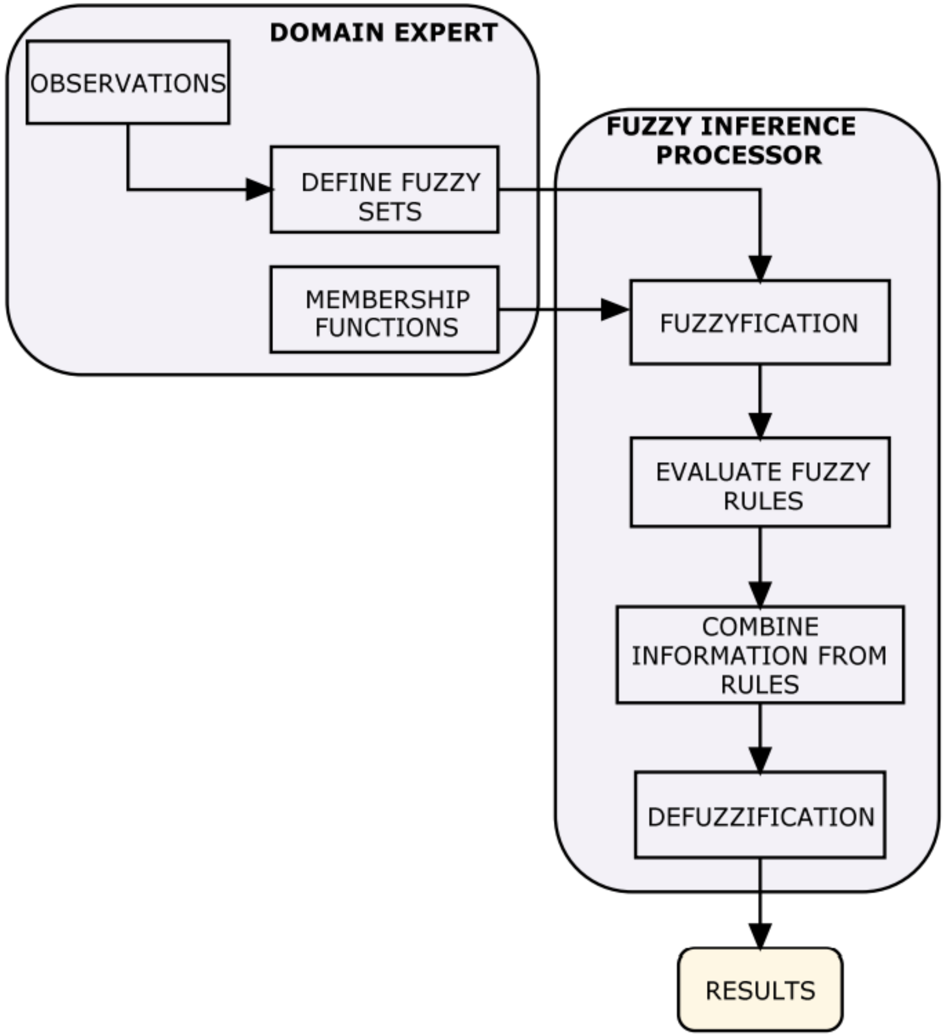
```
fuzzy_rule r3_13f
  if temperature is high
  or current is high
  then current_change becomes reduce.

fuzzy_rule r3_14f
  if temperature is medium
  then current_change becomes no_change.

fuzzy_rule r3_15f
  if temperature is low
  and current is high
  then current_change becomes no_change.

fuzzy_rule r3_16f
  if temperature is low
  and current is low
  then current_change becomes increase.
```

- State variables: `temperature` and `current`
- Action variable: `current_change`
- state variables are the inputs to the fuzzy controller, and the action variable is the output.
- `current_change` has a degree of membership, or possibility

they need to be defuzzified

# Fuzzy Expert Systems

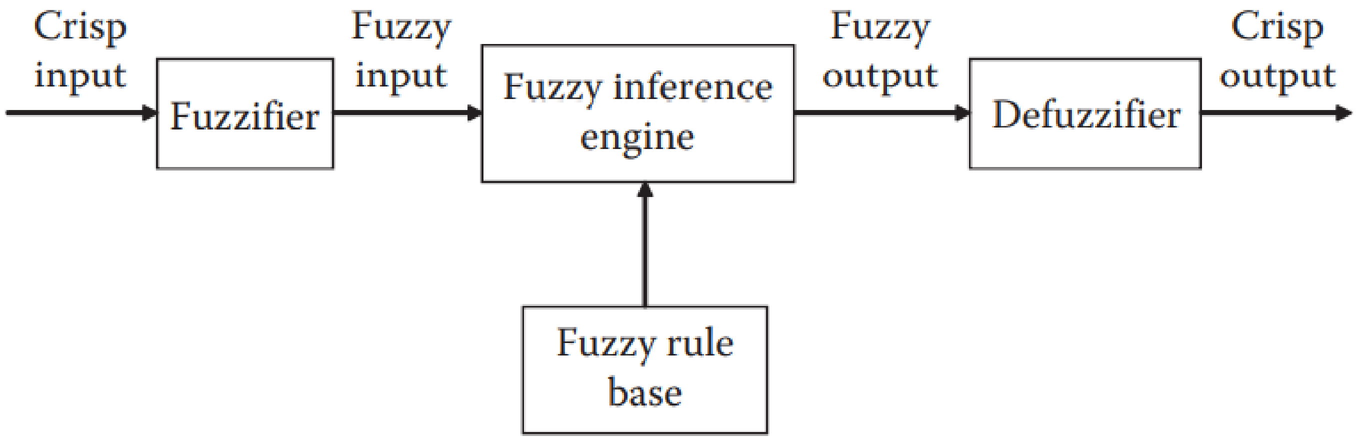
- Steps of Fuzzy Inferencing system
  - Step 1 - Define Fuzzy Sets
  - Step 2 - Relate Observations to Fuzzy Sets
  - Step 3 - Define Fuzzy Rules
  - Step 4 - Evaluate Each Case for all Fuzzy Rules
  - Step 5 - Combine Information from Rules
  - Step 6 - Defuzzify Results




Step 1: Fuzzification
Step 2: Rule Evaluation
Step 3: Defuzzification

# Fuzzy Expert Systems

- Consists of four components: a fuzzifier, a fuzzy rule base, a fuzzy inference engine, and a defuzzifier.




- Fuzzifier converts the numeric input into a linguistic variable - converts a crisp value into a fuzzy singleton within the specified universe of discourse
- Fuzzy rule base is constructed by a collection of fuzzy IF-THEN rules
- Fuzzy inference system – is the kernel, performing fuzzy operations to map from a given input to an output
- Defuzzifier maps the system output from the fuzzy domain into the crisp domain

# Example

Rules in ordinary logic:

- Rule 1: IF temperature is above 50 deg
  OR current is above 10 A
  THEN change motor load by -10%
- Rule 2: IF temperature is between 35 & 50 deg
  THEN no change in motor load
- Rule 3: IF temperature is below 35 deg
  OR current is below 5 A
  THEN change motor load by +10%
- Rule 4: IF temperature is below 50 deg
  OR current is above 10 A
  THEN no change in motor load

Rules in Fuzzy logic:

- Rule 1: IF temperature is high
  OR current is high
  THEN reduce motor load
- Rule 2: IF temperature is medium
  THEN maintain motor load
- Rule 3: IF temperature is low
  AND current is low
  THEN increase motor load
- Rule 4: IF temperature is medium
  AND current is high
  THEN maintain motor load

```python
import numpy as np
import skfuzzy as fuzz
from skfuzzy import control as ctrl

Temp = ctrl.Antecedent(np.arange(0, 101, 1), 'Temperature')
Current = ctrl.Antecedent(np.arange(0, 17, 1), 'Current')
Load = ctrl.Consequent(np.arange(-10, 11, 1), 'Load')

names = ['low', 'medium', 'high']
Temp.automf(names = names)
Temp.view()
Current.automf(names = names)
Current.view()
Load['Reduce'] = fuzz.trimf(Load.universe, [-10, -5, 0])
Load['Maintain'] = fuzz.trimf(Load.universe, [-1, 0, 1])
Load['Increase'] = fuzz.trimf(Load.universe, [0, 5, 10])
Load.view()

rule1 = ctrl.Rule(Temp['high'] | Current['high'], Load['Reduce'])
rule2 = ctrl.Rule(Temp['medium'], Load['Maintain'])
rule3 = ctrl.Rule(Temp['low'] | Current['low'], Load['Increase'])
rule4 = ctrl.Rule(Temp['medium'] & Current['medium'], Load['Maintain'])

Loading_ctrl = ctrl.ControlSystem([rule1, rule2, rule3, rule4])
Loading = ctrl.ControlSystemSimulation(Loading_ctrl)

Loading.input['Temperature'] = 35
Loading.input['Current'] = 12

Loading.compute()
print (Loading.output['Load'])
Load.view(sim=Loading)
```