Expert Systems and Fuzzy Logic

November 2024

COMP8811 Data Analytics and Intelligence

Expert Systems Content

- Basics
- Applications
- Development
- Structure
- Inferencing Methods

Basics

- **What is an Expert system?**
  - A computer program that simulates the thought process of a human expert to ***solve complex decision problems*** in a specific domain.

Basics

- **An expert system is for operating as an interactive system that**
  - **responds** to questions
  - **asks** for clarifications
  - **makes** recommendations
  - **aids** the decision making process.

- ***Examples***
  - Computer diagnosis and troubleshooting
  - medical diagnosis
  - delicate medical surgery

Basics

- **How an expert system operates?**
  - It uses both ***facts and heuristics*** to solve difficult decision problems based knowledge acquired from an expert.
- **Some example…**
  - **Knowledge based system:** https://www.youtube.com/watch?v=MxI_tzOs6ME
- Have you tried ***NHS expert systems***? Discover how its operates and explain it in one short paragraph.
  - https://www.youtube.com/watch?v=yPI7tLGhXgo

Basics : Why do we need it?

Because of limitations associated with ***conventional human decision-making processes:***

- Humans expertise is very limited
- Humans get tired form physical or mental work load.
- Humans have limited working memory.
- Humans are unable to comprehend large amount of data quickly.
- Humans are slow in recalling information stored in memory.
- Humans are subject to deliberate or inadvertent in their action.
- Humans can deliberately avoid decision responsibilities.
- Humans lie, hide and die.

Basics : Why we need it?

Because of limitation associated with *conventional* ***programming and traditional decision-making support:***

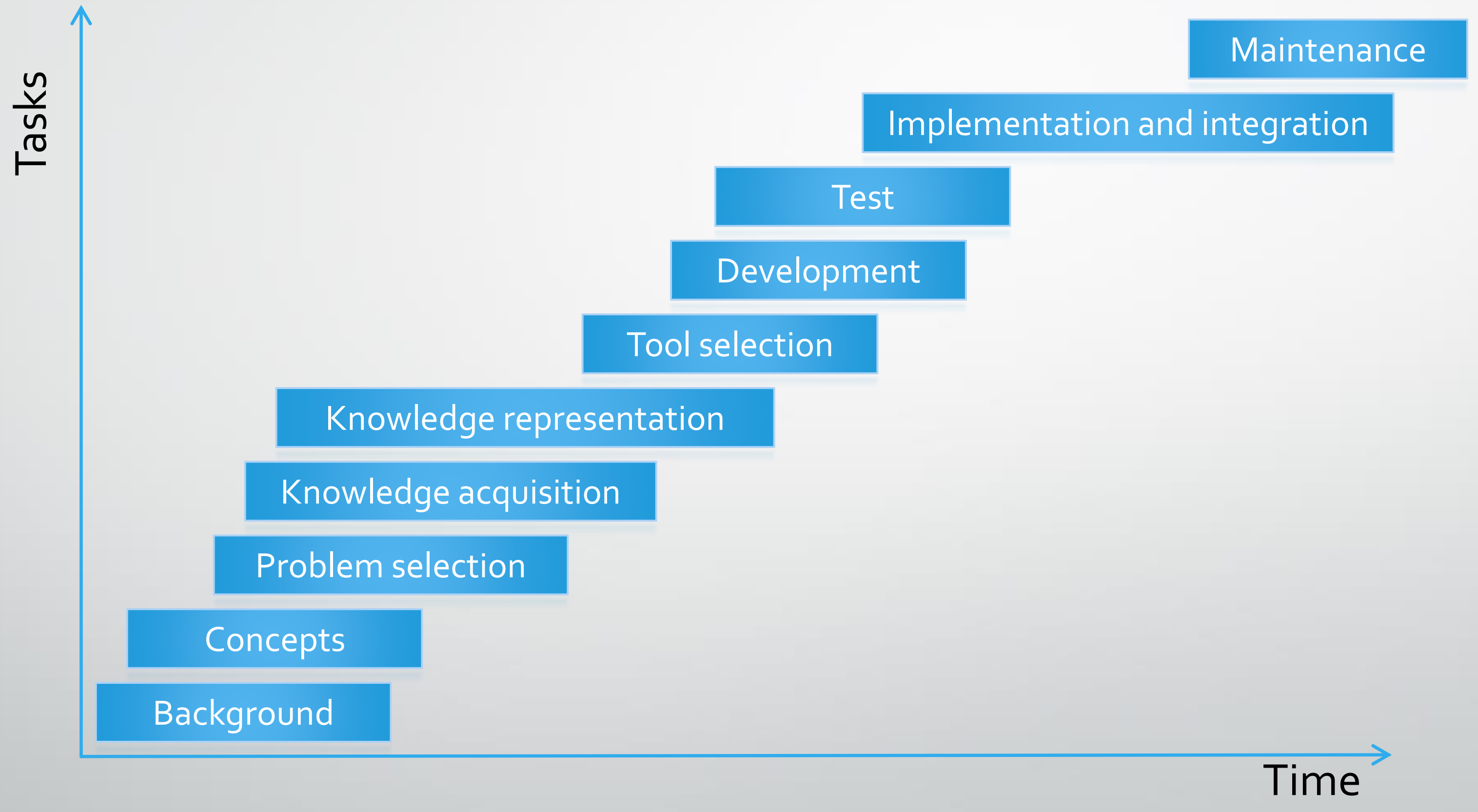
- They are algorithmic in nature and depend only on raw machine power.
- They do not make use of the effective heuristic approaches used by human expert.
- They are not easily adoptable to changing problem environment.
- They seek explicit and factual solutions that may not be possible.

Applications

What are main applications of expert systems?

- Interpreting and identifying
- Predicting
- Diagnosing
- Designing and planning
- Monitoring
- Debugging and testing
- Instructing and training
- Controlling

Expert System Development



Hierarchy of expert systems development process

Expert System Development

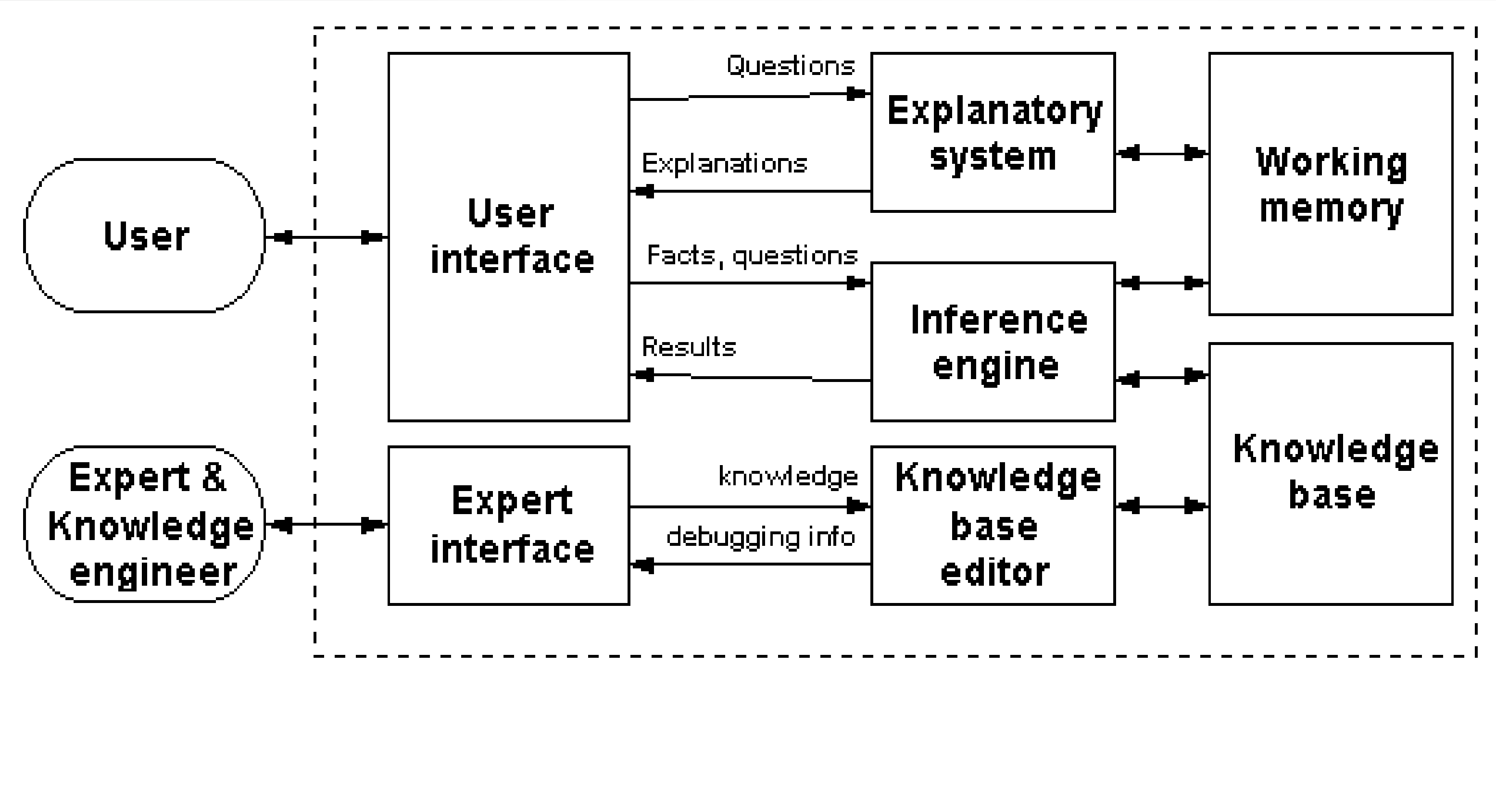
- What is a successful expert system?
  - The one with making decision process similar to that of human brain.
- How human brain makes a decision?
  - By using mixture of factual and heuristic knowledge.
- What is heuristic knowledge?
  - It is composed of intuition, judgment and logical inference. It is an indisputable strength of human.
- Lets answer the first question more accurately now:
  - Successful expert systems are those combine facts and heuristics and thus merge human knowledge with computer power in solving problems.

Expert Systems Structure

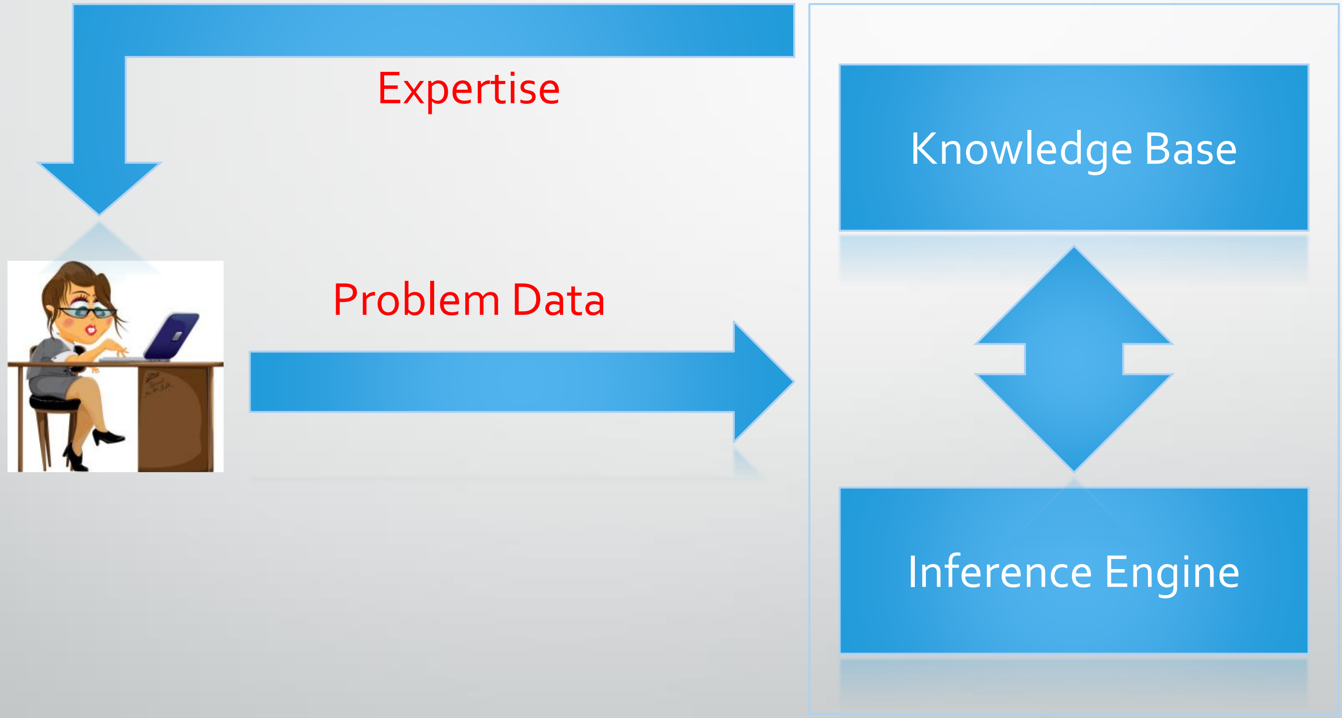
Expert systems contain 3 main components:

- Knowledge Base consists of problem solving rules, procedures, and relevant data.
- Working Memory refers to task-specific data for the problem under consideration.
- Inference Engine is a generic control mechanism that applies the knowledge base to the task-specific data to arrive at some solutions or conclusion.

Expert Systems Structure



Expert Systems Structure



Expert System = Knowledge Base + Inference Engine

Expert Systems Structure

Human experts or consulting firm

Knowledge Engineers → Knowledge Base

User

→ Problem Data

Commercial vendor e.g., VP-Expert

→ Inference Engine

Expert Systems Structure

Human experts or consulting firm

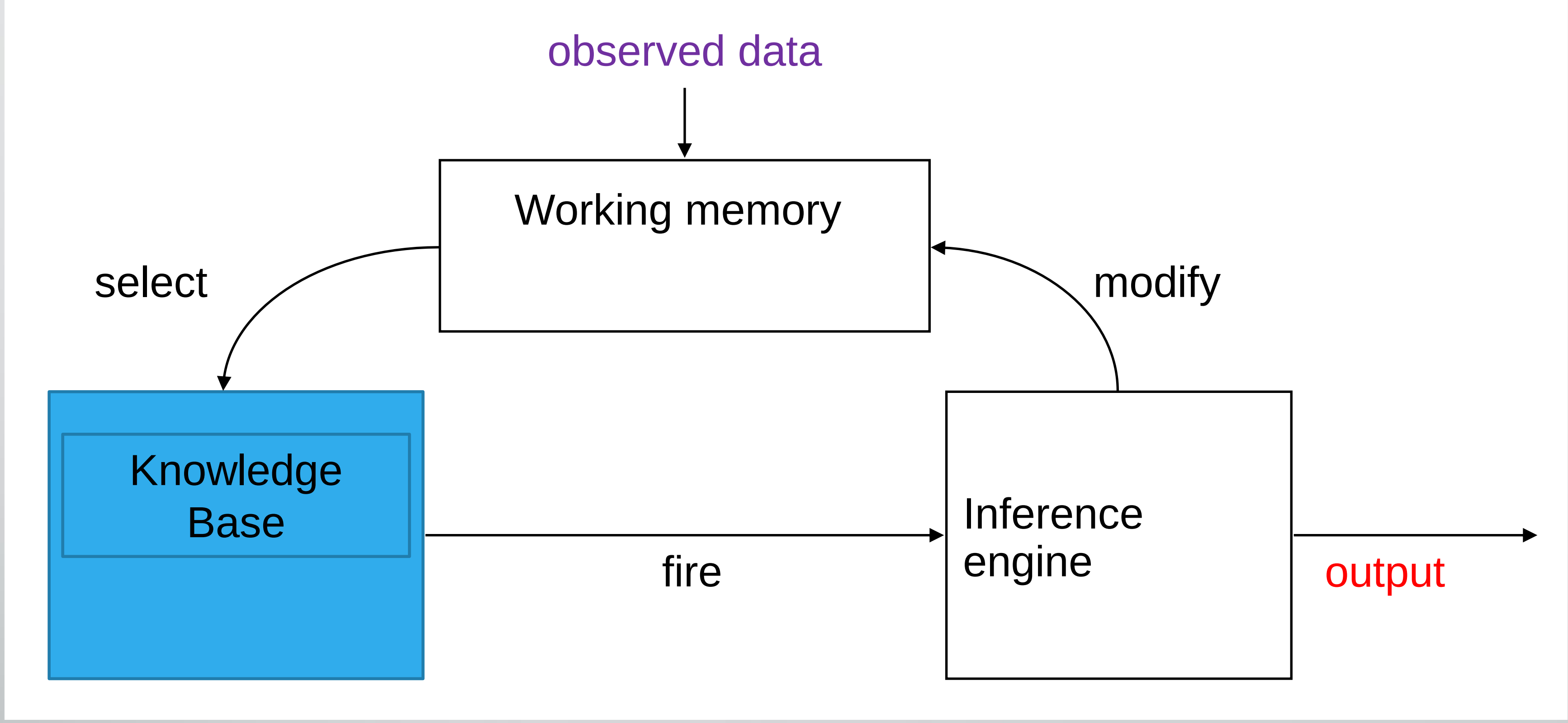
Knowledge Base Structure

- It constitutes **problem solving rules**, **facts or intuitions** that a human expert might use in solving problems in a given problem domain.
- It is usually stored in terms of **if-then rules**.

Example

- **IF** the patient has headache **THEN** ask how long he/she has suffered this pain.

Expert Systems Structure



Expert Systems Structure

How to create the knowledge base:

- The **knowledge engineer** establishes a dialog with the human expert to elicit knowledge.
- The **knowledge engineer** codes the knowledge explicitly in the knowledge base.
- The **expert** evaluates the expert system and gives a critique to the knowledge engineer.

Expert Systems Structure

Try ***NHS online expert system*** for an imaginary scenario.

- Discover the if-then rules for the chosen scenario.

https://111.wales.nhs.uk/selfassessments/?locale=en&term=A

Inferencing Methods

General methods of inference:

- **Forward Reasoning (data-driven)**
  - reasoning **from facts to the conclusions** resulting from those facts
  - best for **prognosis (prediction)**, **monitoring**, and **control**.

- **Backward Reasoning (query-driven)**
  - reasoning in **reverse from a hypothesis**, a potential conclusion to be proved to the facts that support the hypothesis
  - best for **diagnosis problems**.

Inferencing Methods : Forward reasoning

Example for Knowledge Base Rules:

1. IF (Symptom1, X) AND (Symptom2, X) THEN ADD (Allergy, X)
2. IF (Season, Spring) THEN ADD (Symptom1, John)
3. IF (Season, Spring) THEN ADD (Symptom2, John)
4. IF (Allergy, X) OR (Asthma, X) THEN ADD (Inhaler Needed, X)
5. IF (Inhaler Needed, X) THEN DEL (Happy X)
6. IF (Symptom1, X) THEN DEL (Swimming, X)

Inferencing Methods : Forward reasoning

- Assume that initially we have a working memory of **Patient: John**

(Season, Spring)

(Happy John)

(Swimming, John)

Knowledge Base Rules

1. **IF** (Symptom1, X) AND (Symptom2, X) **THEN** ADD (Allergy, X)
2. **IF (Season, Spring) THEN ADD (Symptom1, John)**
3. **IF** (Season, Spring) **THEN** ADD (Symptom2, John)
4. **IF** (Allergy, X) OR (Asthma, X) **THEN** ADD (Inhaler Needed, X)
5. **IF** (Inhaler Needed, X) **THEN** DEL (Happy X)
6. **IF** (Symptom1, X) **THEN** DEL (Swimming, X)

Step 1:

(Season, Spring)

(Happy John)

(Swimming, John)

2 → (Symptom1, John)

3 → X

Inferencing Methods : Forward reasoning

Updated Working Memory (after Step 1)

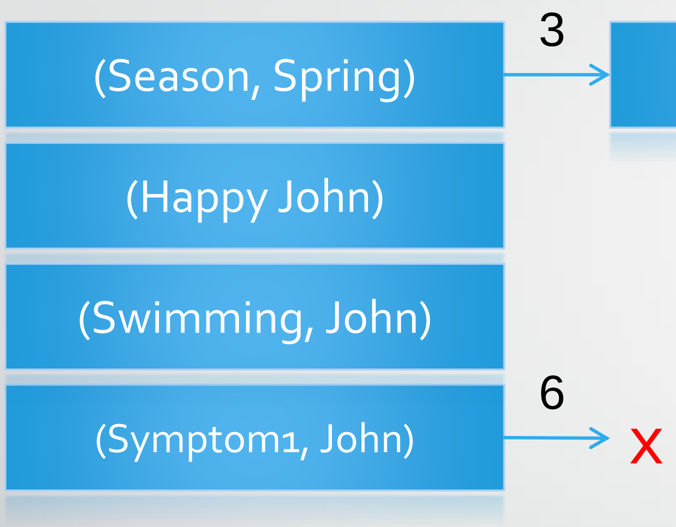
- (Season, Spring)
- (Happy John)
- (Swimming, John)

→

- (Season, Spring)
- (Happy John)
- (Swimming, John)
- (Symptom1, John)

Inferencing Methods : Forward reasoning

Step 2



Knowledge Base Rules

① **IF** (Symptom1, X) AND (Symptom2, X) **THEN** ADD (Allergy, X)

② **IF** (Season, Spring) **THEN** ADD (Symptom1, John)

③ **IF (Season, Spring) THEN ADD (Symptom2, John)**

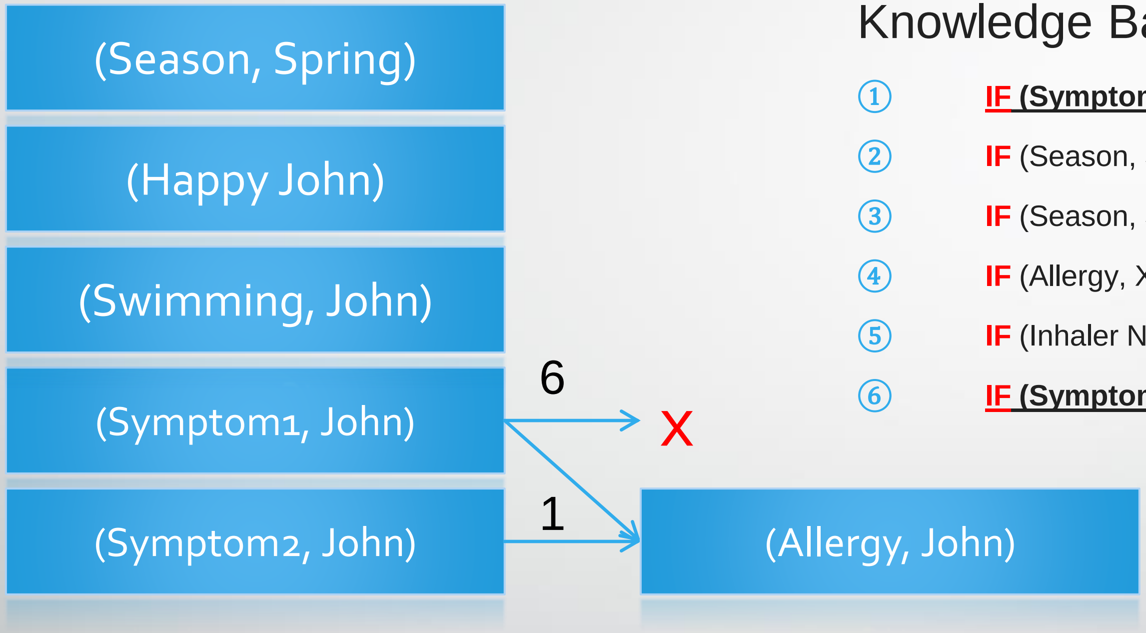
④ **IF** (Allergy, X) OR (Asthma, X) **THEN** ADD (Inhaler Needed, X)

⑤ **IF** (Inhaler Needed, X) **THEN** DEL (Happy X)

⑥ **IF (Symptom1, X) THEN DEL (Swimming, X)**

Inferencing Methods : Forward reasoning

Step 3



Knowledge Base Rules

① **IF** (Symptom1, X) AND (Symptom2, X) **THEN** ADD (Allergy, X)

② **IF** (Season, Spring) **THEN** ADD (Symptom1, John)

③ **IF** (Season, Spring) **THEN** ADD (Symptom2, John)

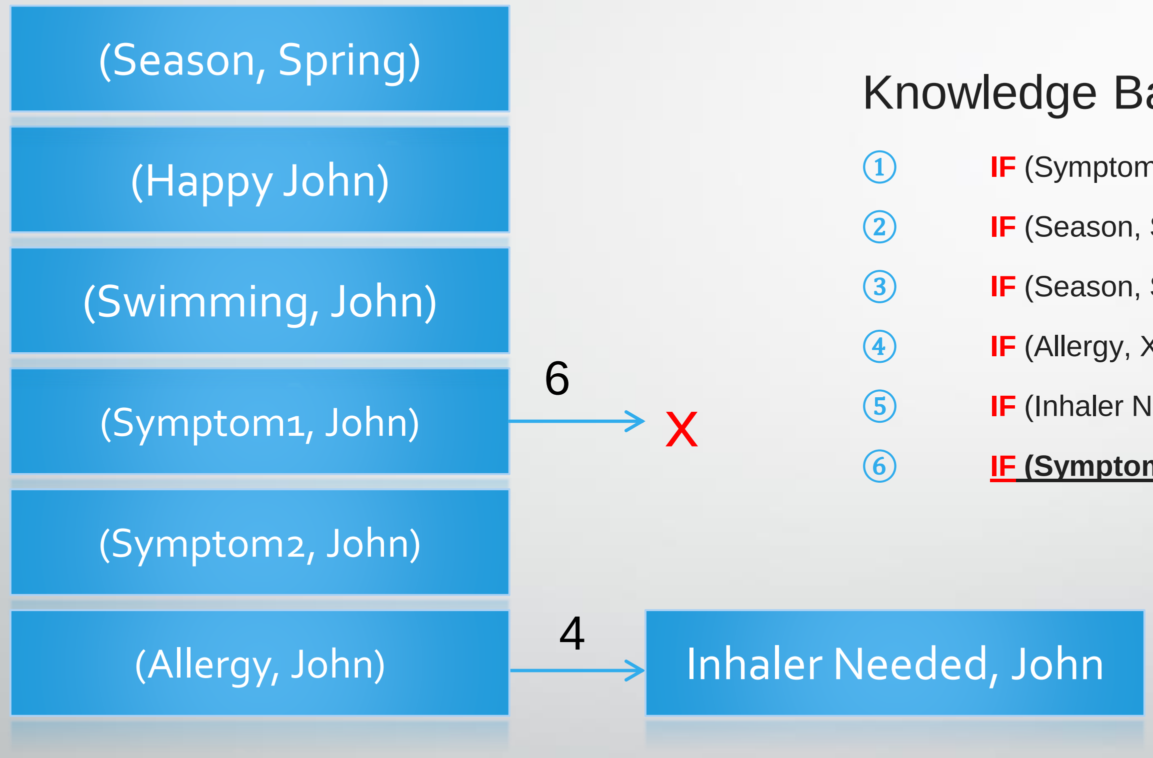
④ **IF** (Allergy, X) OR (Asthma, X) **THEN** ADD (Inhaler Needed, X)

⑤ **IF** (Inhaler Needed, X) **THEN** DEL (Happy X)

⑥ **IF** (Symptom1, X) **THEN** DEL (Swimming, X)

Inferencing Methods : Forward reasoning

Step 4

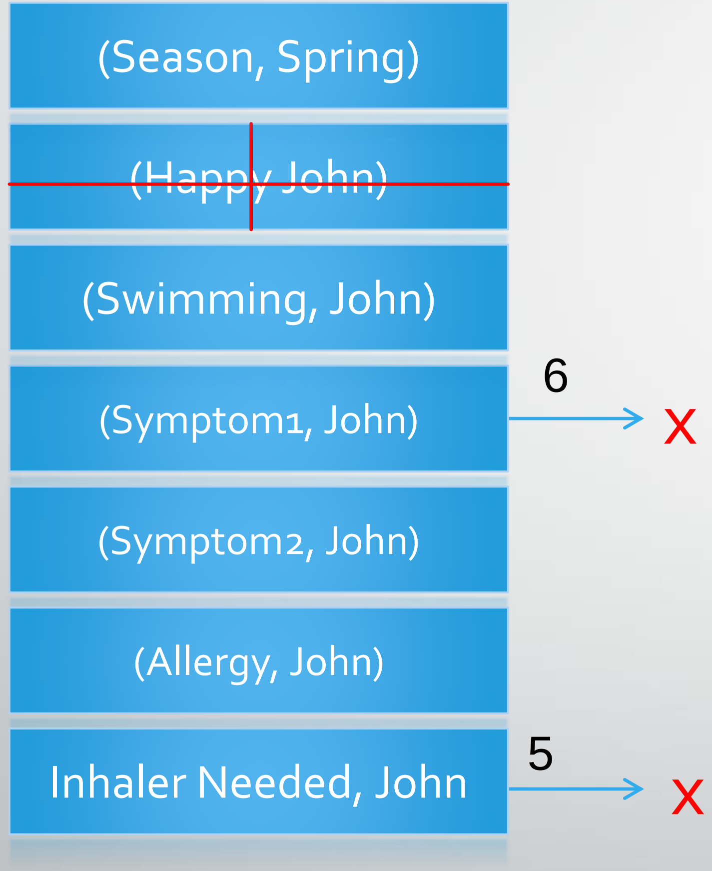


Knowledge Base Rules

1. **IF** (Symptom1, X) AND (Symptom2, X) **THEN** ADD (Allergy, X)
2. **IF** (Season, Spring) **THEN** ADD (Symptom1, John)
3. **IF** (Season, Spring) **THEN** ADD (Symptom2, John)
4. **IF** (Allergy, X) OR (Asthma, X) **THEN** ADD (Inhaler Needed, X)
5. **IF** (Inhaler Needed, X) **THEN** DEL (Happy X)
6. **IF (Symptom1, X) THEN DEL (Swimming, X)**

Inferencing Methods : Forward reasoning

Step 5



Knowledge Base Rules

1. **IF** (Symptom1, X) AND (Symptom2, X) **THEN** ADD (Allergy, X)
2. **IF** (Season, Spring) **THEN** ADD (Symptom1, John)
3. **IF** (Season, Spring) **THEN** ADD (Symptom2, John)
4. **IF** (Allergy, X) OR (Asthma, X) **THEN** ADD (Inhaler Needed, X)
5. **IF** (Inhaler Needed, X) **THEN** DEL (Happy X)
6. **IF (Symptom1, X) THEN DEL (Swimming, X)**

Inferencing Methods : Forward reasoning

Step 6

- (Season, Spring)
- ~~(Swimming, John)~~
- (Symptom1, John) ← 6
- (Symptom2, John)
- (Allergy, John)
- Inhaler Needed, John

Knowledge Base Rules

① **IF** (Symptom1, X) AND (Symptom2, X) **THEN** ADD (Allergy, X)

② **IF** (Season, Spring) **THEN** ADD (Symptom1, John)

③ **IF** (Season, Spring) **THEN** ADD (Symptom2, John)

④ **IF** (Allergy, X) OR (Asthma, X) **THEN** ADD (Inhaler Needed, X)

⑤ **IF** (Inhaler Needed, X) **THEN** DEL (Happy X)

⑥ **IF (Symptom1, X) THEN DEL (Swimming, X)**

Inferencing Methods : Forward reasoning

Final Working Memory

- (Season, Spring)
- (Symptom1, John)
- (Symptom2, John)
- (Allergy, John)
- Inhaler Needed, John

Inferencing Methods : Backward reasoning

- **Backward Reasoning (query-driven)**
  - reasoning in **reverse from a hypothesis**, a potential conclusion to be proved to the facts that support the hypothesis
  - best for **diagnosis problems**.

- **Example for** Knowledge Base Rules:

① IF (Symptom1, X) AND (Symptom2, X) THEN ADD (Allergy, X)

② IF (Season, Spring) THEN ADD (Symptom1, John)

③ IF (Season, Spring) THEN ADD (Symptom2, John)

④ IF (Allergy, X) THEN ADD (Inhaler Needed, X)

⑤ IF (Asthma, X) THEN ADD (Inhaler Needed, X)

⑥ IF (Season, Spring) THEN ADD (Weather, Rainy)

IF (Year, 2010) THEN ADD (Economy, Bad)

Inferencing Methods : Backward reasoning

- Let us assume that initially we have a user facts set with **Patient: John**

(Season, Spring)

(Year, 2012)

- We want to prove that **John needed an Inhaler** that time.

Inferencing Methods : Backward reasoning

Step 1

(Season, Spring)

(Year, 2012)

① IF (Symptom1, X) AND (Symptom2, X) THEN ADD (Allergy, X)

② IF (Season, Spring) THEN ADD (Symptom1, John)

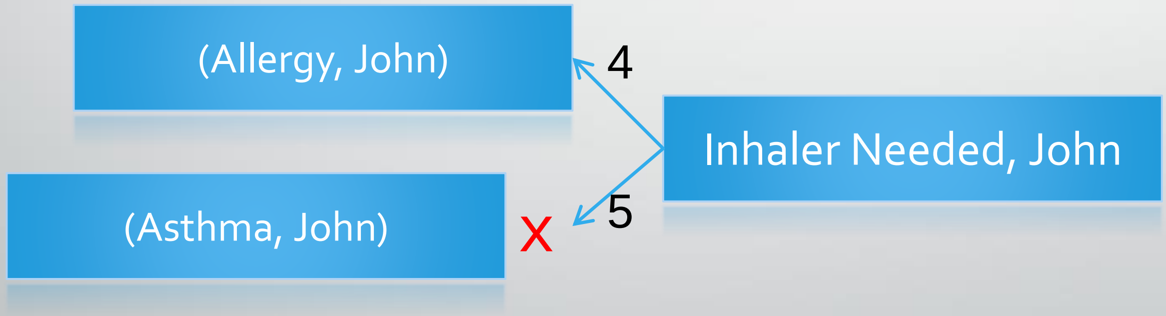
③ IF (Season, Spring) THEN ADD (Symptom2, John)

④ IF (Allergy, X) THEN ADD (Inhaler Needed, X)

⑤ IF (Asthma, X) THEN ADD (Inhaler Needed, X)

⑥ IF (Season, Spring) THEN ADD (Weather, Rainy)

⑦ IF (Year, 2010) THEN ADD (Economy, Bad)



Inferencing Methods : Backward reasoning

Step 2: (we don't need update the working memory)

(Season, Spring)

(Year, 2012)

① IF (Symptom1, X) AND (Symptom2, X) THEN ADD (Allergy, X)

② IF (Season, Spring) THEN ADD (Symptom1, John)

③ IF (Season, Spring) THEN ADD (Symptom2, John)

④ IF (Allergy, X) THEN ADD (Inhaler Needed, X)

⑤ IF (Asthma, X) THEN ADD (Inhaler Needed, X)

⑥ IF (Season, Spring) THEN ADD (Weather, Rainy)

⑦ IF (Year, 2010) THEN ADD (Economy, Bad)

(Symptom2, John)

(Symptom1, John)

1

(Allergy, John)

Inferencing Methods : Backward reasoning

- **Step 3:**

(Season, Spring)

(Year, 2012)

① IF (Symptom1, X) AND (Symptom2, X) THEN ADD (Allergy, X)

② IF (Season, Spring) THEN ADD (Symptom1, John)

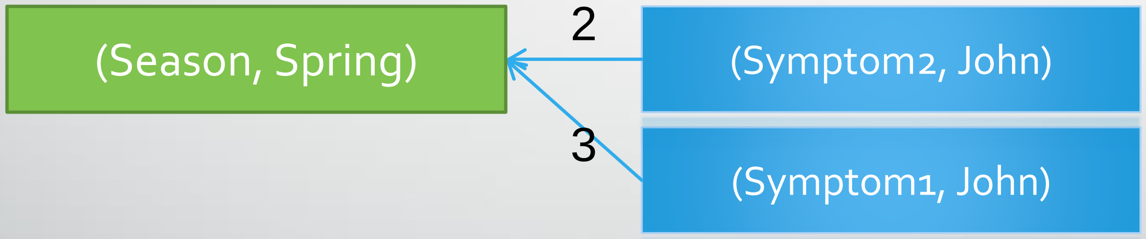
③ IF (Season, Spring) THEN ADD (Symptom2, John)

④ IF (Allergy, X) THEN ADD (Inhaler Needed, X)

⑤ IF (Asthma, X) THEN ADD (Inhaler Needed, X)

⑥ IF (Season, Spring) THEN ADD (Weather, Rainy)

⑦ IF (Year, 2010) THEN ADD (Economy, Bad)



- **(Season, Spring)** is among *the initial facts*
  - so it is proven that **(Inhaler Need, John)**

Inferencing Methods

Forward Reasoning vs. Backward Reasoning

- **Forward Reasoning** may be better if you have ***lots of things*** you want to prove :
  - A small set of initial facts
  - Lots of different rules allow you to draw the same conclusion

- **Backward Reasoning** may be better if you are trying to prove a ***single fact***
  - Given a large set of initial facts forward reasoning can fire lots of rules can be in any cycle.

Advantages of Expert Systems

- Provide consistent answers for repetitive decisions, processes and tasks.
- Hold and maintain significant levels of information.
- Reduce employee training costs
- *Centralize* the decision making process.
- Create efficiencies and reduce the time needed to solve problems.

Advantages of Expert Systems (cont.)

- Combine multiple human expert intelligences
- Reduce the amount of human errors.
- Give strategic and comparative advantages creating entry barriers to competitors
- Review transactions that human experts may overlook.

Disadvantages of Expert Systems

- **Lack** human common sense needed in some decision making.
- Will **not** be able to give the *creative responses* that human experts can give in unusual circumstances.
- Domain experts **cannot** always clearly explain their logic and reasoning.
- **Challenges** of automating complex processes.
- **Lack** of flexibility and ability to adapt to changing environments.
- **Not** being able to recognize when no answer is available.

Break

- 15 min

Class Activity

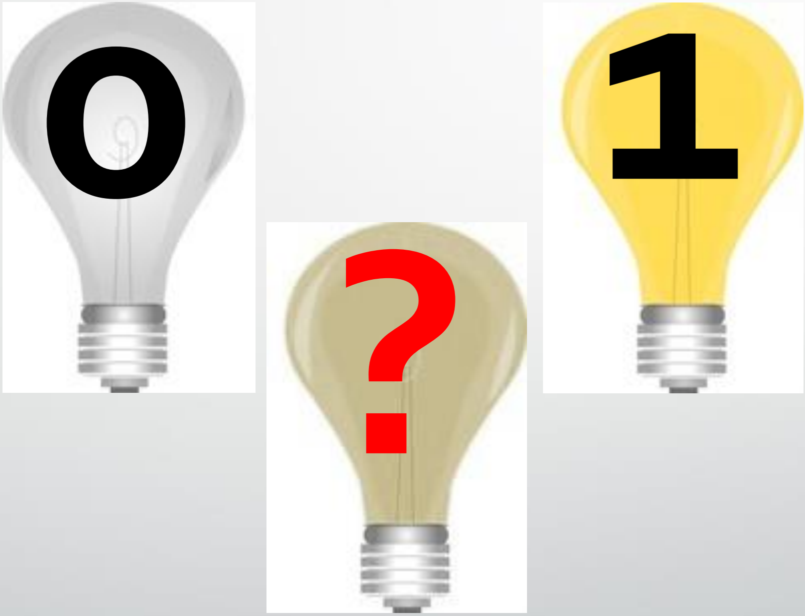
- Form a group of 3-4 students
- Discuss the current AI tools in market, pick and example and explore:
    1. How does the tool work?
    2. What are the advantages that tool can offer education system?
    3. What are the disadvantages of the tool for education system?
    4. If we could set policies to use the tools in an effective way, what would be those policies
- Present to the class

Fuzzy Logic

Content

- Basics and Concepts
- Membership Functions
- Fuzzy Rules
- Fuzzifcation and Defuzzification
- Fuzzy Operations
- Fuzzy Systems Design

Basics and Definitions



Basics and Definitions

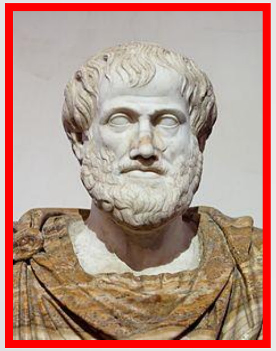
0

?

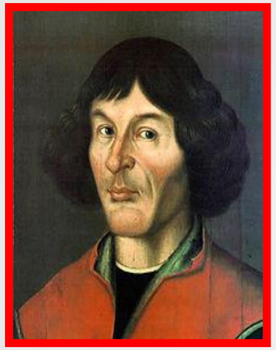
?

1

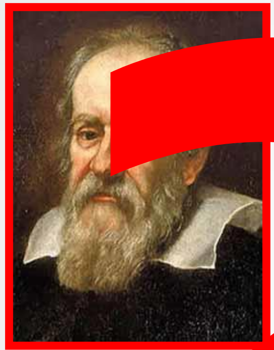
Basics and Definitions



Aristotle
384–322 BC



Copernic
1473-1543



Galileo
1564-164

Boole
1815-1864

Shannon
1916-2001

Mauchly
1907-1980

Basics and Definitions

Aristotle logic of “true or flase” was accepted by the Greek scholars and later got spread all over Europe; first by the Romans and then through different religions. It matches the idea of God and Evil or Heaven and Hell.

Basics and Definitions

- **The Origin of Fuzzy Logic**

500 BC (200 years before Aristotle): Buddha established a philosophy in India. Buddha philosophy was based on the thought that the world is filled with contradictions, that almost everything contains some of its opposite, or in other words, that an statement can be true a and not-true at the same time.

Lotfi A Zadeh

Basics and Definitions

- In 1964, prof. Zadeh (an Iranian Mathematician) started wondering, if there wasn't a better logic to use in computing systems.
- He had the idea that if you could tell an air-conditioner to work a little faster when it gets hotter, or similar problems, it would be much more efficient than having to give a rule for each temperature.
- That was the day fuzzy logic the way we know it today was born; with fuzzy logic you can tell an air-conditioner to slow down as soon as it gets chilly.

Basics and Definitions

- Crisp sets
- Fuzzy sets
- Fuzzy logic
- Linguistic variables
- Membership function
- Membership value (degree)
- Fuzzy Rule
- Fuzzy Operations

Basics and Definitions

- **Crisp Sets**

  Elements of a crisp set represent ***precise quantities***.

  - For a crisp set of **real numbers**: elements are, for example, 1, -1, 0.0232, 3.98343
  - For a crisp set of **binary numbers**: elements are either 1 or 0
  - For a crisp set of **logical values**: elements are either True or False

Basics and Definitions

Crisp Logic

A ⇒ B

A: "Student" is motivated.

B: "Student" passes ABI.

What if John is only somewhat motivated?

John is motivated ⇒ John passes ABI

to what extent? **grade?**

Basics and Definitions

- **Fuzzy Sets**

Elements of a fuzzy set represent the (membership) degree to which a ***quality*** is possessed.

Simple ***Fuzzy Variables*** have membership values in the range of [0,1].

- For example, 2.5 is half 2 and half 3.

Basics and Definitions

Fuzzy Logic

A ⇒ B

A: "Student" is motivated.

B: "Student" passes ABI.

What if John is only somewhat greedy?

John is motivated ⇒ John passes ABI

Membership degree = 0.7 **Grade=70%**

Basics and Definitions

Linguistic Variables

Fuzzy set can formulate ***linguistic variables set*** efficiently.

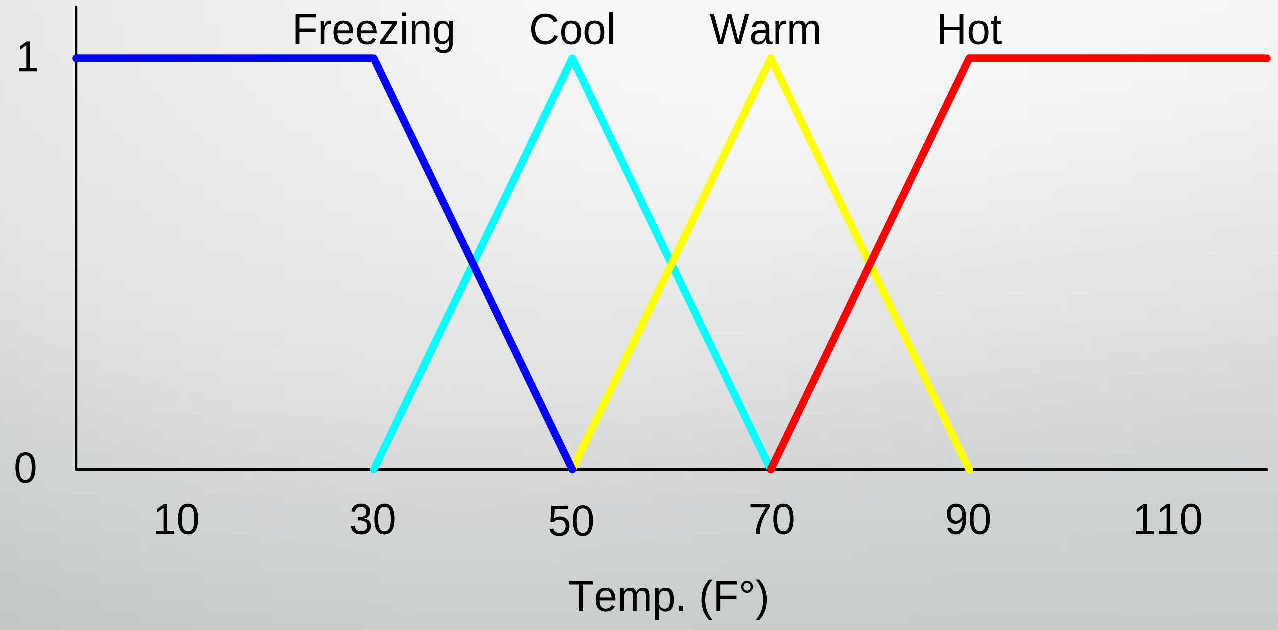
For example, the linguistic variable '**Temp**' can take the following values:

Temp: {Freezing, Cool, Warm, Hot}

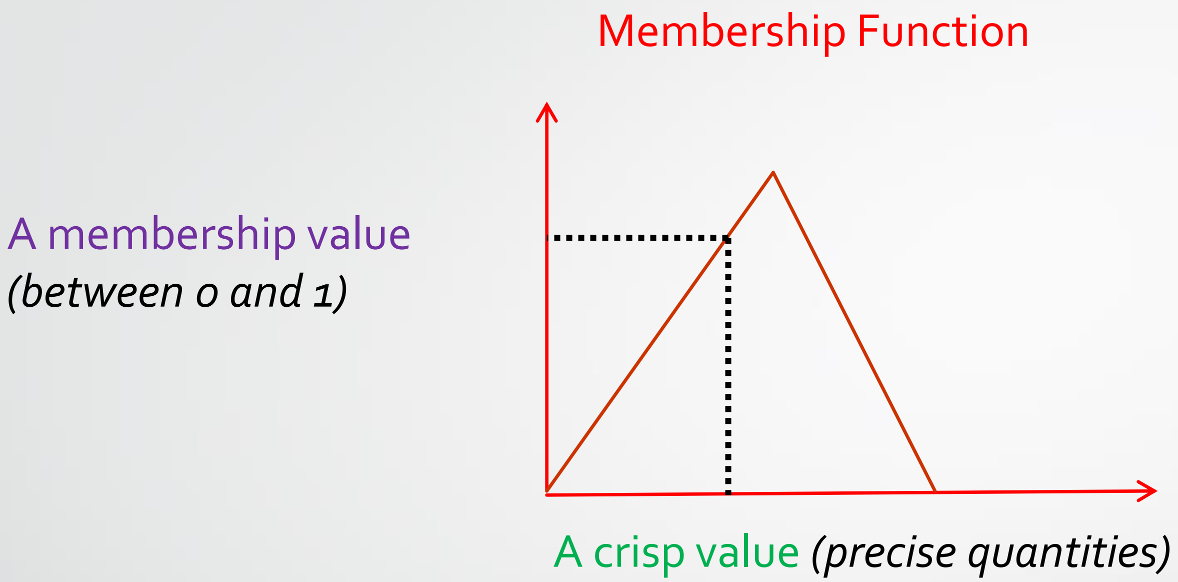
- Example..
  - Q: What is the temperature?
  - A: It is 80^o F.
  - Q: How does it feel?
  - A: Hot.
  - Q: How hot is it?
  - A: ?

Membership Functions

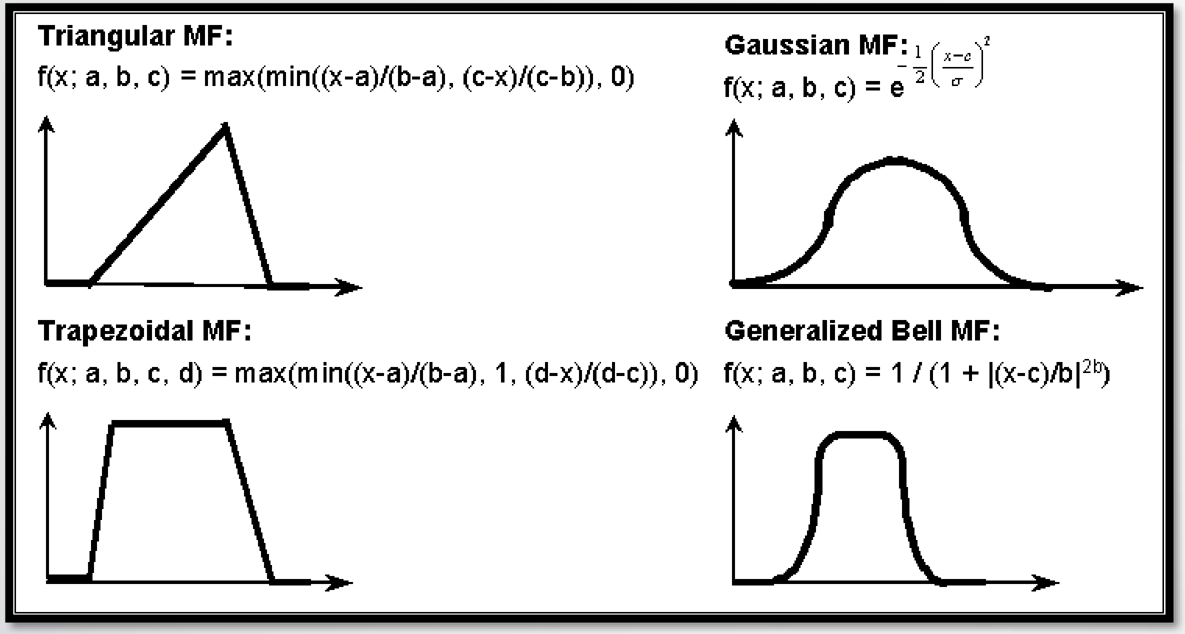
- **Membership functions** represent the ***degree of truth*** as an extension of evaluation.
- For example membership function for “**Temp**” values can be defined as:

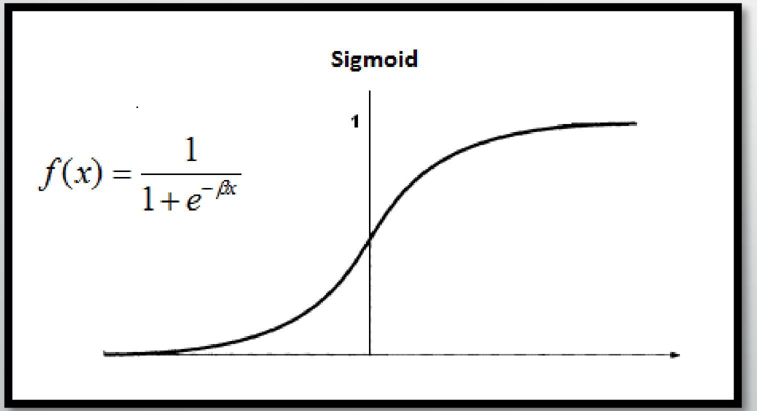


Membership Functions



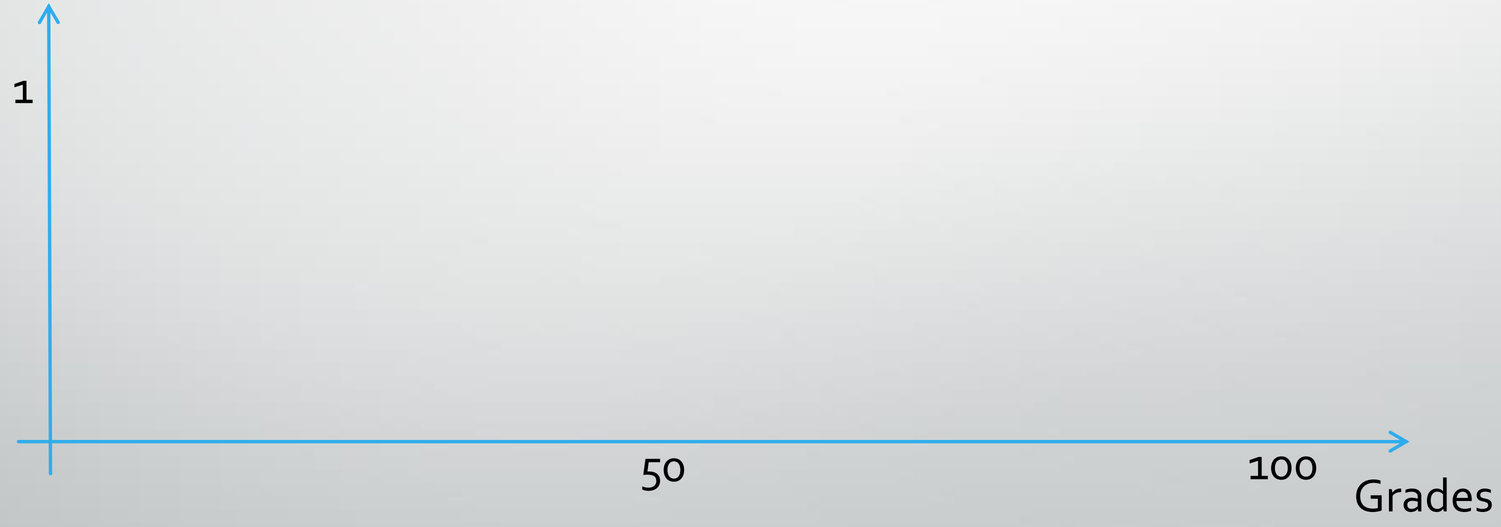
Membership Functions





Membership Functions

1. Create a membership function for the following fuzzy variable.
2. Evaluate yourself! How good student you are (membership degree)?
   - **Student** = {**Excellent**, **Good**, **Medium**, **Bad**, **Vey Bad**}



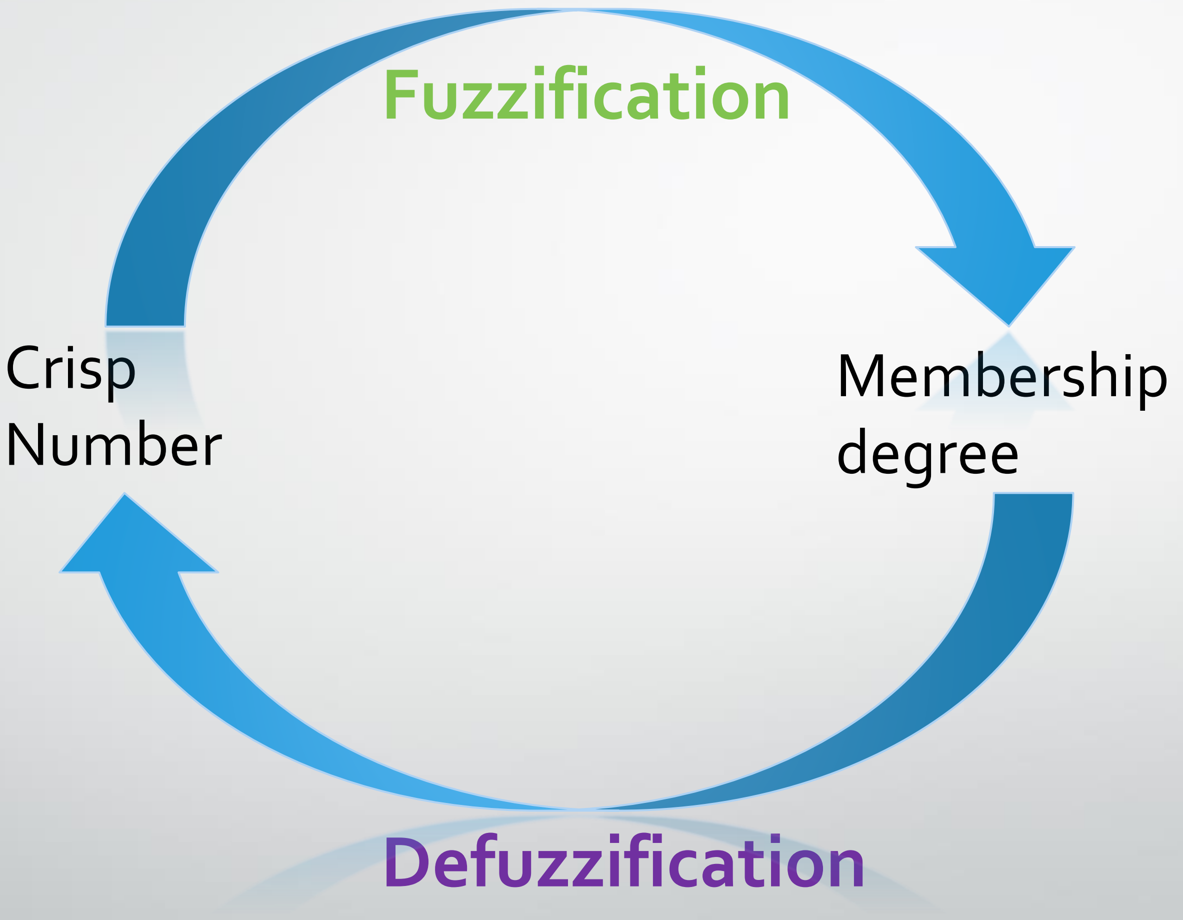
Fuzzy Rules

- **A fuzzy rule** is defined as a conditional statement in the form:
  - **IF** x~A **THEN** y~B
  - where **x** and **y** are linguistic variables (values determined by fuzzy)

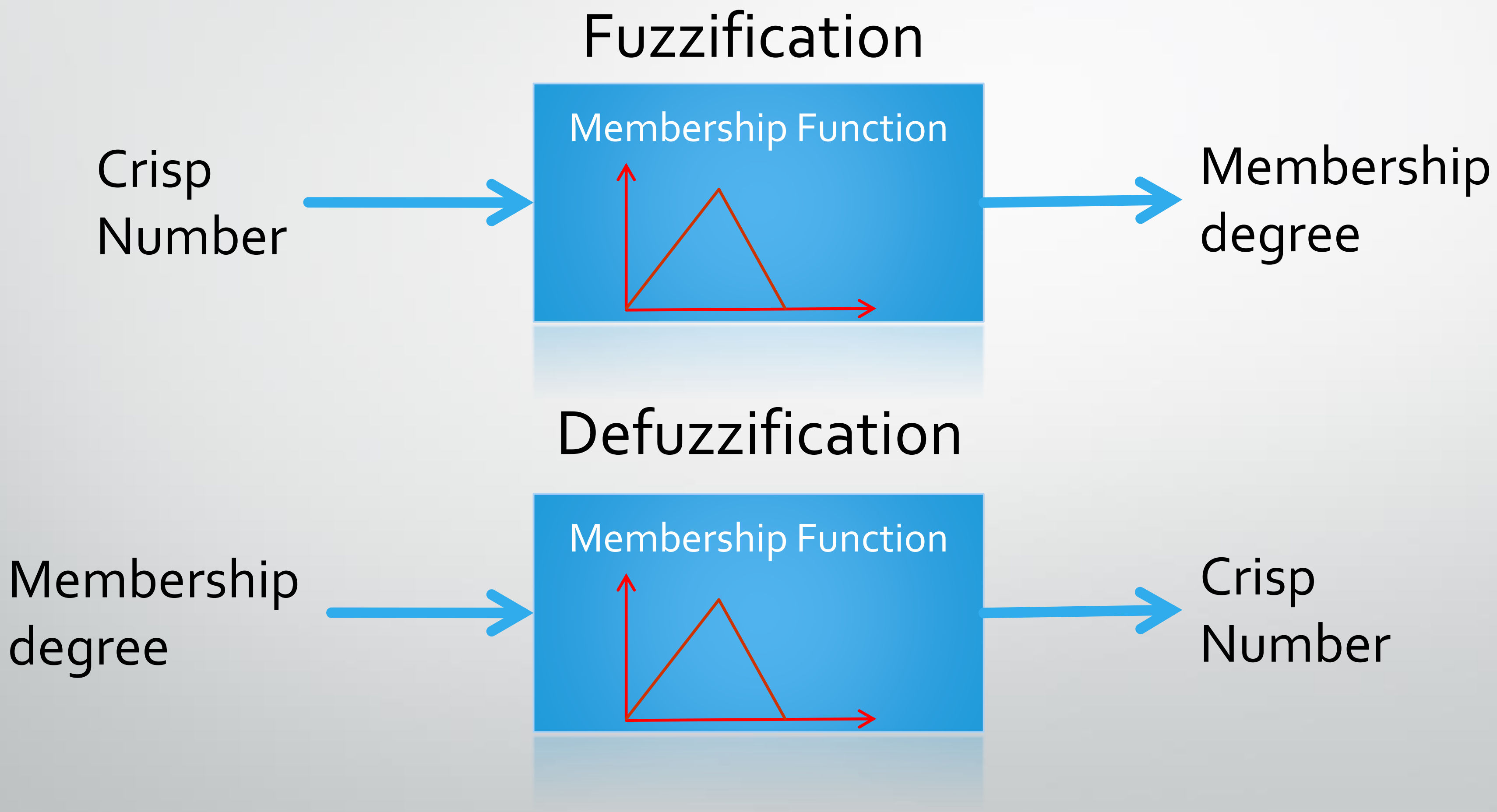
- **An example**
  - **IF** Weather is Sunny **THEN** Driving Speed is Fast.

From a fuzzy set (membership function associated)

Fuzzification and Defuzzification

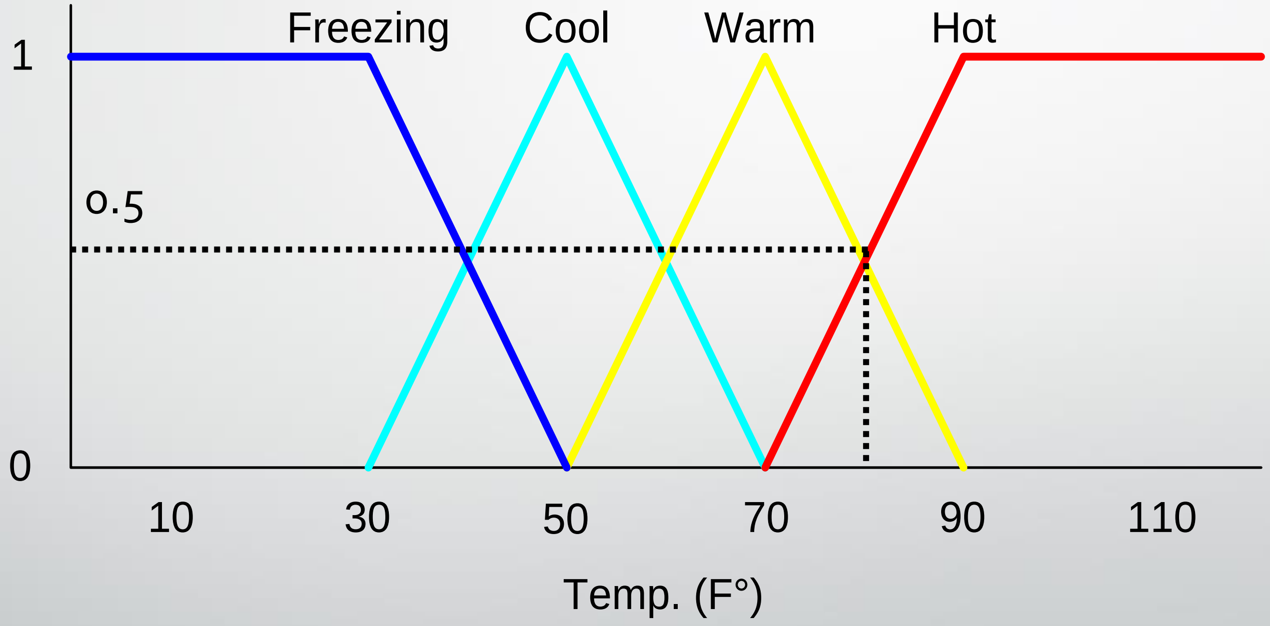


Fuzzification and Defuzzification



Fuzzification and Defuzzification

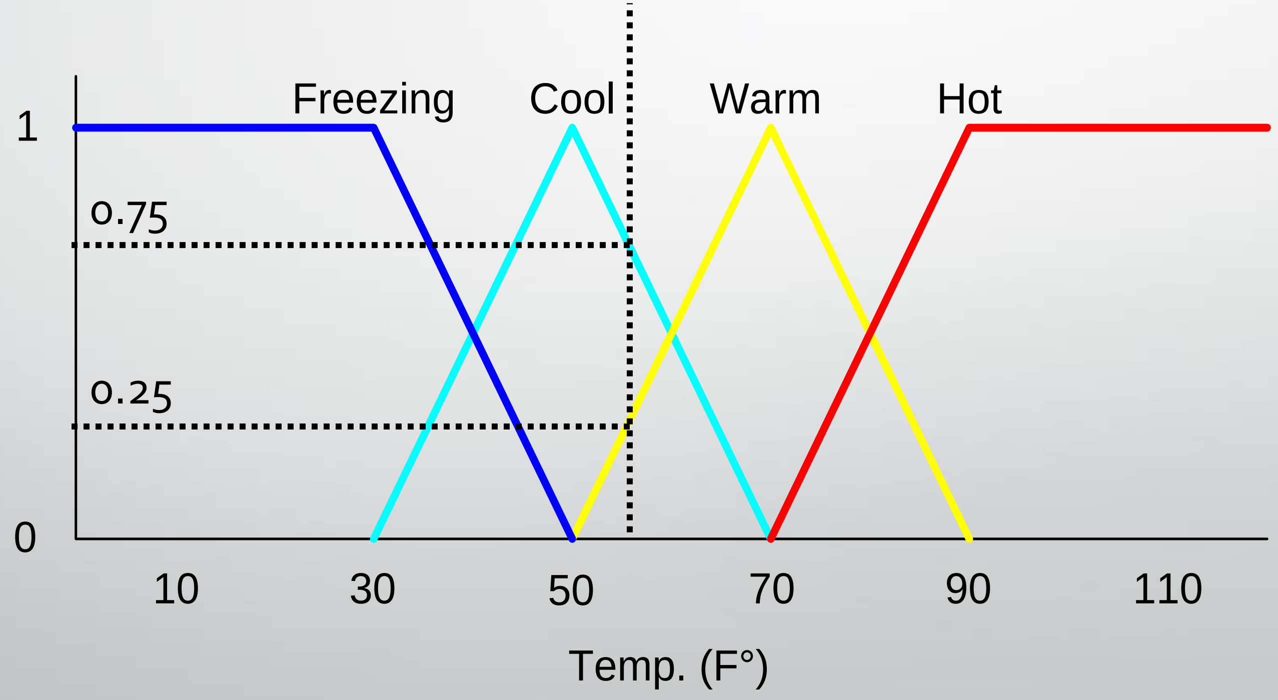
- **Example… How hot is it (80° F) ?**



Fuzzification

- Q: How does the weather feel **in 55° F**?

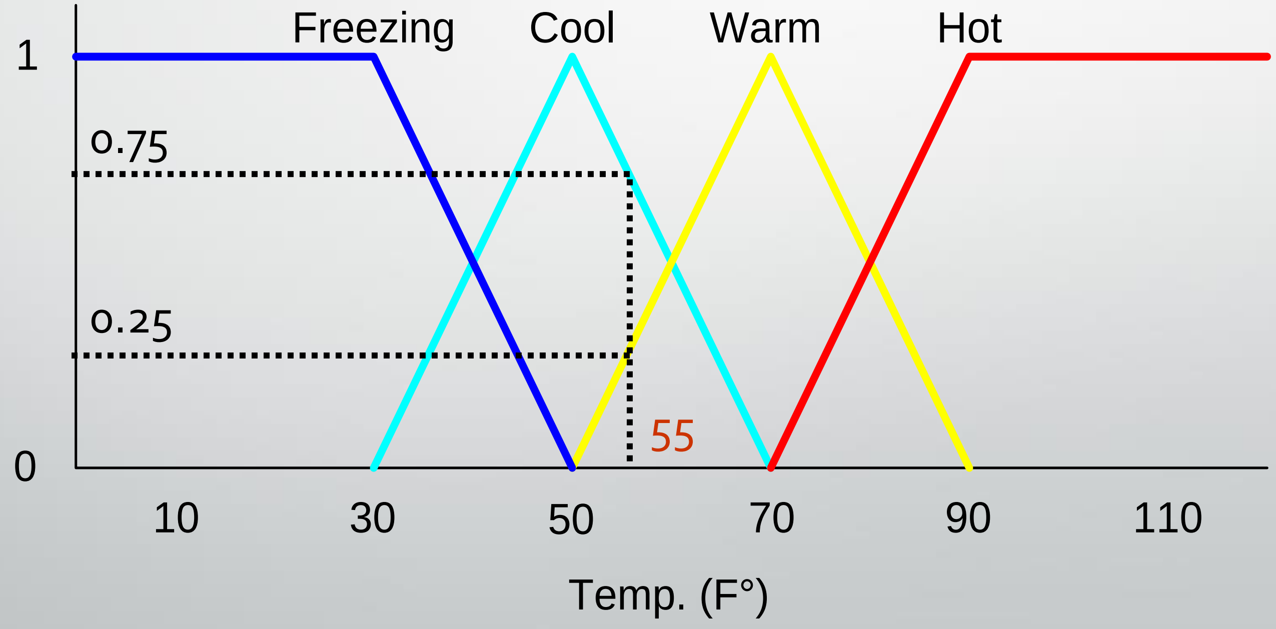
A: It is **25% Warm** and **75% Cool**.



Defuzzification

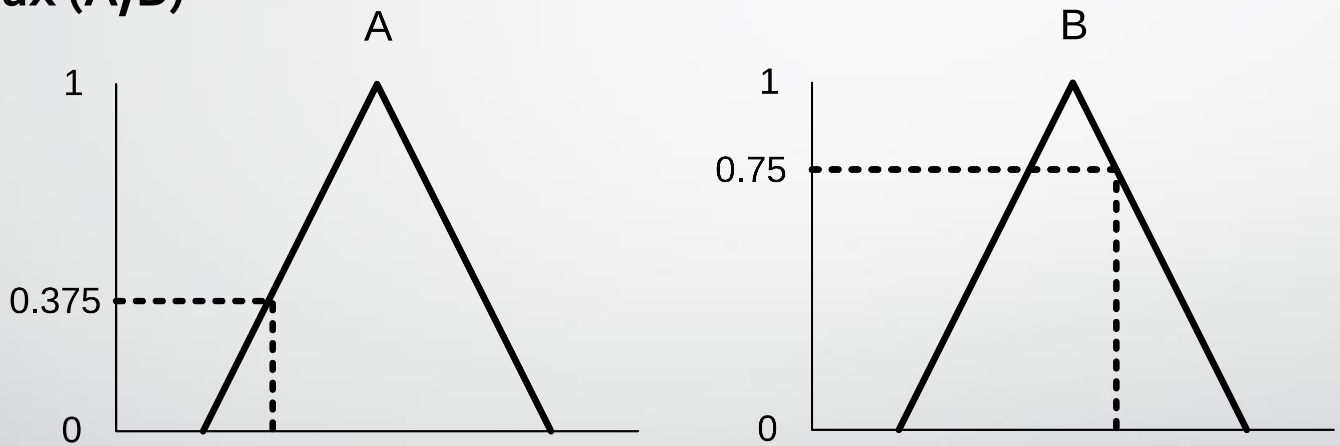
- Q: It feels **25% Warm** and **75% Cool.** What is the **temperature**?

55



Fuzzy Operations

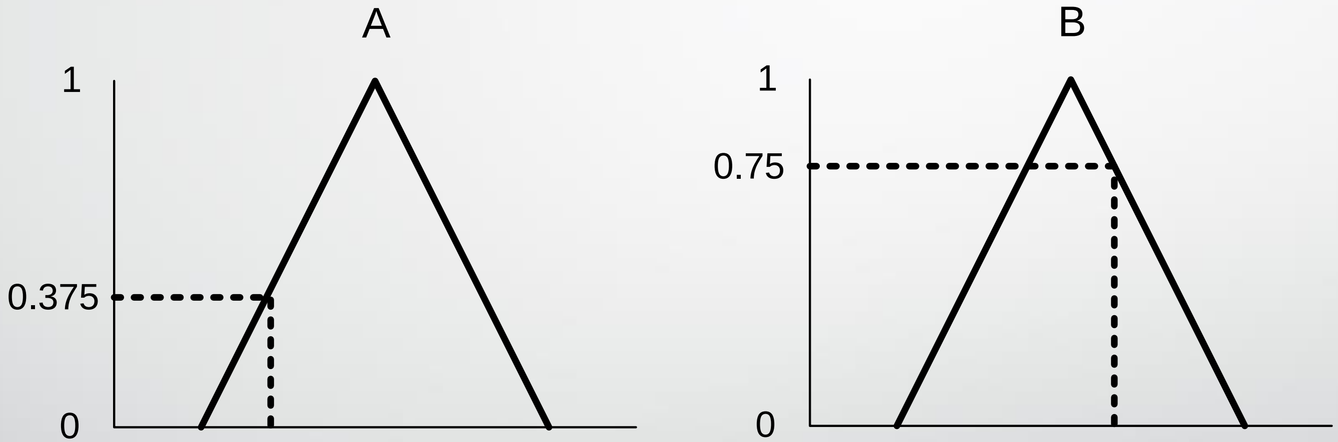
- **Fuzzy Disjunction (OR)**
  - **A∨B = max (A,B)**



$$(A \vee B = C) \Rightarrow (C = 0.75)$$

Fuzzy Operations

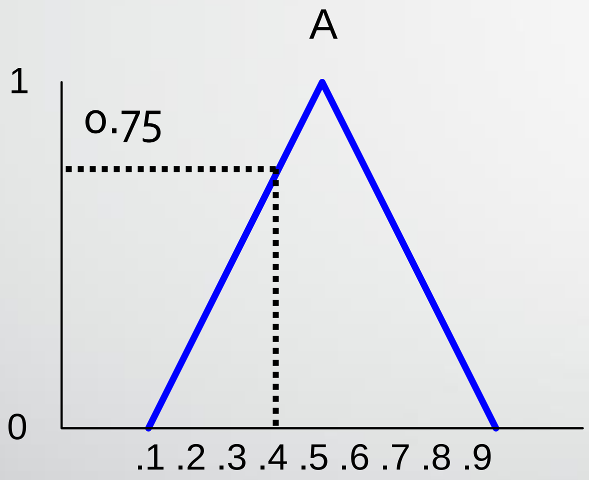
- **Fuzzy Conjunction (AND)**
  - **A∧B = min (A,B)**

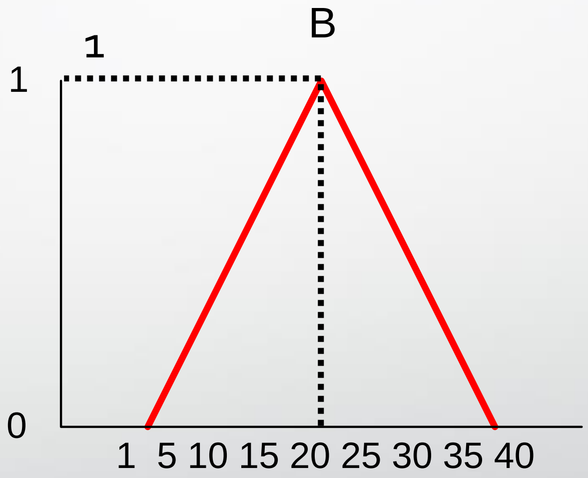


(A∧B = C) ⇒ (C = 0.375)

Fuzzy Operations

- Calculate **A∧B** and **A∨B** if A=0.4 and B=20?





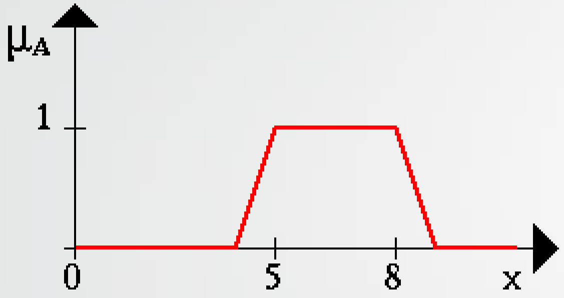
A∧B = 0.75

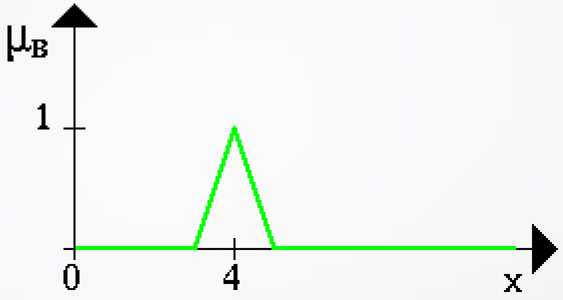
A∨B = 1.00

Fuzzy Operations

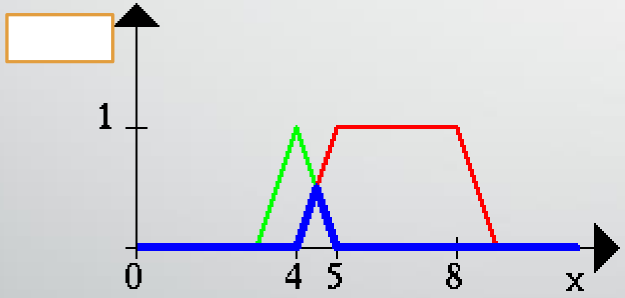
A∨B = max (A,B)

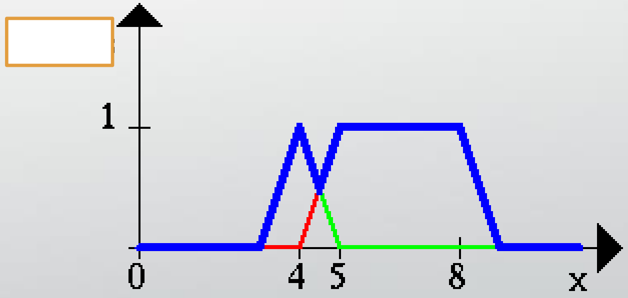
A∧B = min (A,B)





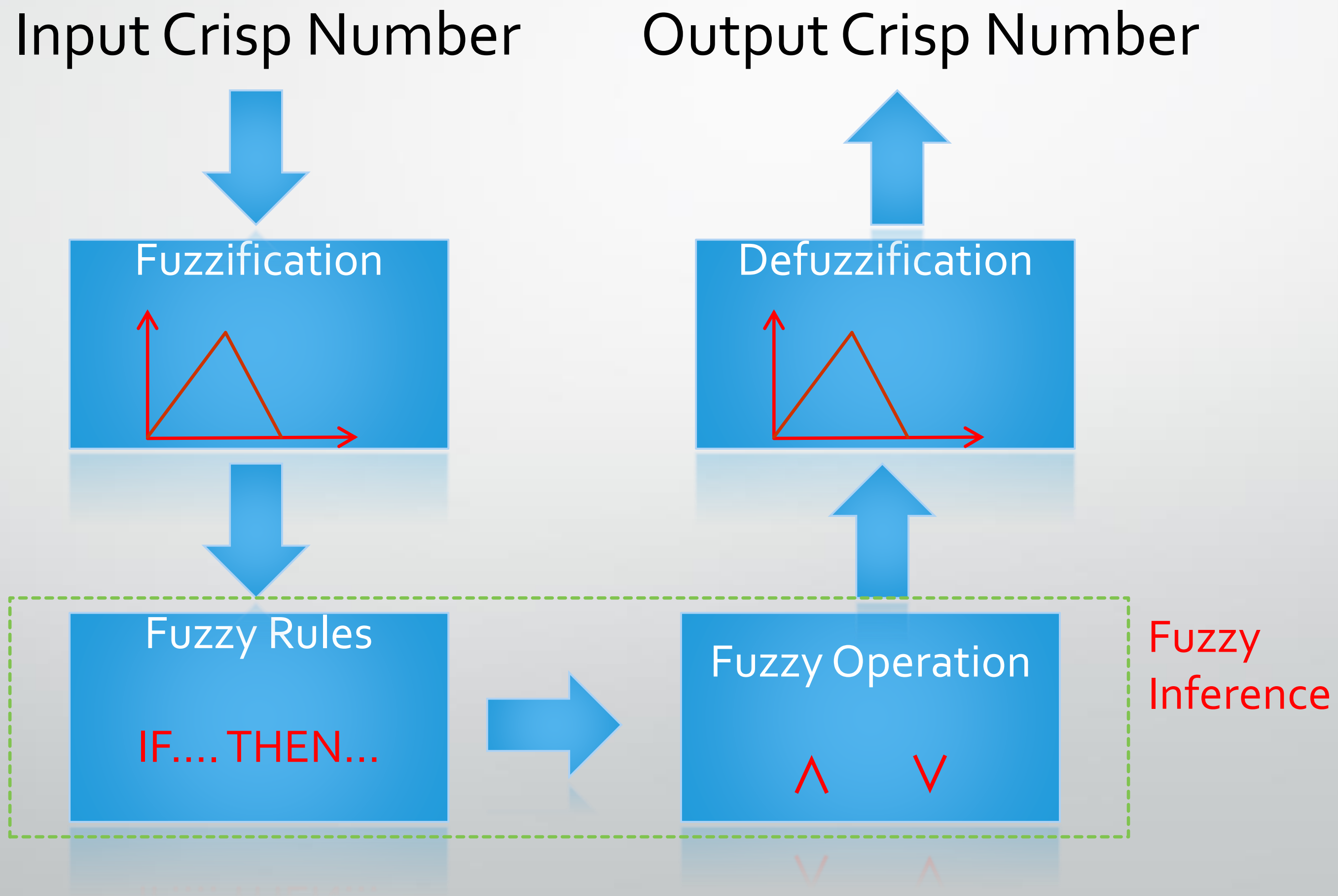
- **AND vs OR**





Fuzzy Systems

- **Mamdani Fuzzy Systems Model**



Fuzzy Systems

Fuzzy Systems Design

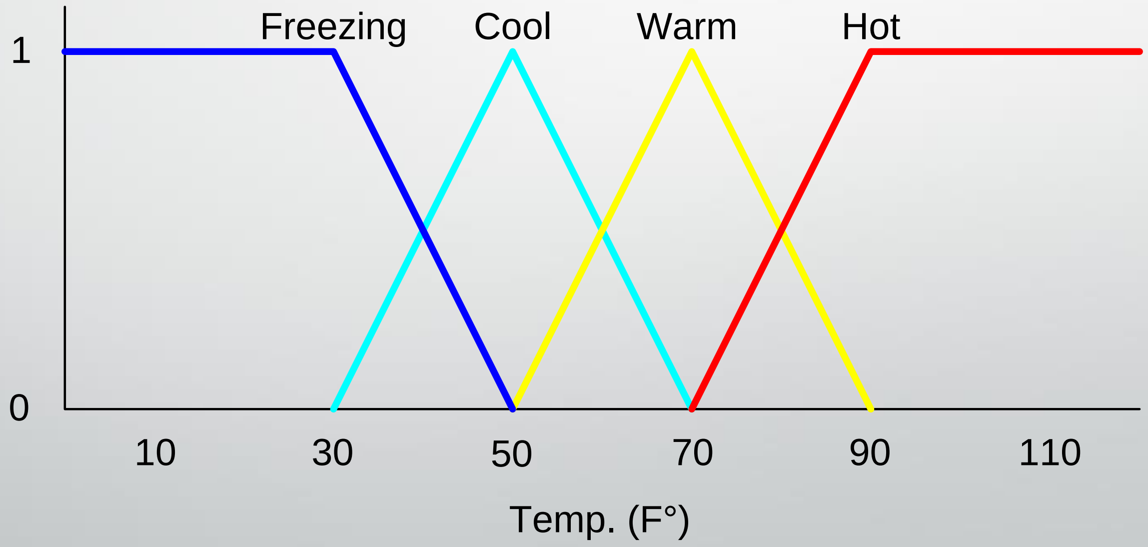
- **Step 1**: Determining linguistic rules
- **Step 2**: Developing fuzzy rules
- **Step 3**: Developing membership function
- **Step 4**: Implementation

Fuzzy Systems Design : Example

- **Q: How fast can I drive today?**
  - **Based on:** Cloud cover, Temperature, and Speed
  - **Step 1**: Defining Linguistic Rules
    - ❑ If it's **Sunny** and **Warm**, drive **Fast**
    - ❑ If it's **Cloudy** and **Cool**, drive **Slow**
  - **Step 2**: Defining Fuzzy Rules
    - ❑ $Sunny(Cover) \wedge Warm(Temp) \Rightarrow Fast(Speed)$
    - ❑ $Cloudy(Cover) \wedge Cool(Temp) \Rightarrow Slow(Speed)$

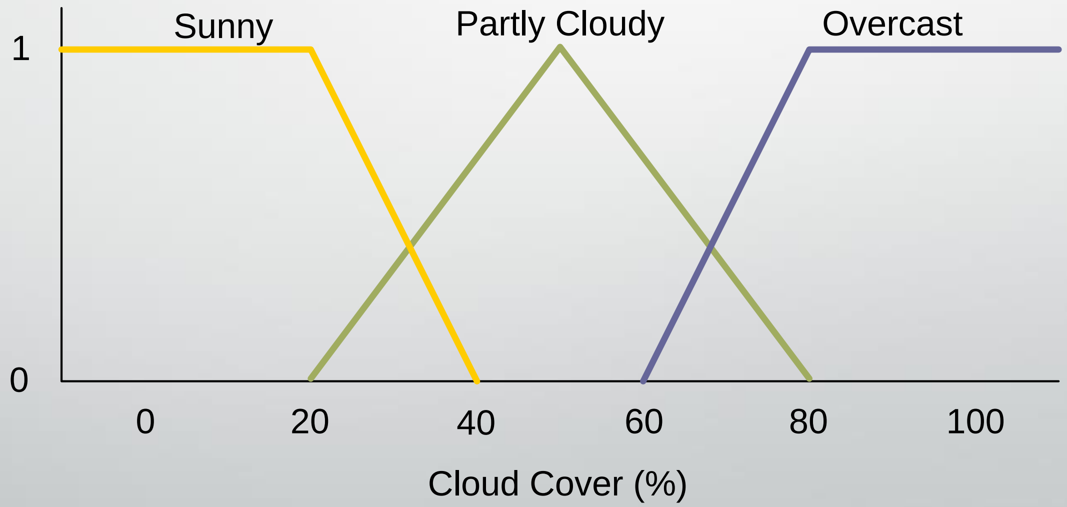
Fuzzy Systems Design : Example

- **Q: How fast can I drive today?**
  - **Step 3: Defining Input Membership Functions (1/3)**
    - **Temp={ Freezing, Cool, Warm, Hot }**



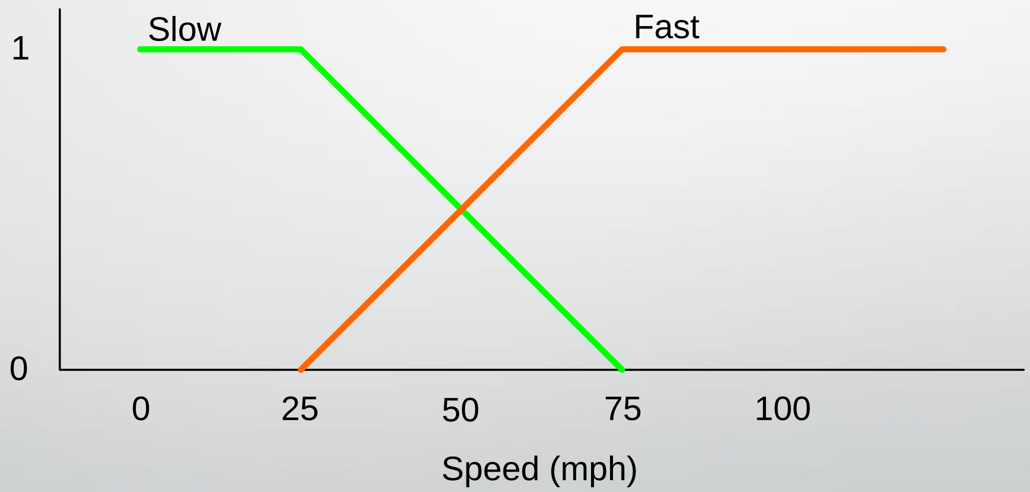
Fuzzy Systems Design : Example

- Q: How fast can I drive today?
  - Step 3: Defining Input Membership Functions (2/3)
    - Cloud Cover={ Sunny, Partly Cloudy, Overcast }



Fuzzy Systems Design : Example

- **Q: How fast can I drive today?**
  - **Step 3: Defining Output Membership Functions (3/3)**
    - **Speed = {Slow, Fast}**

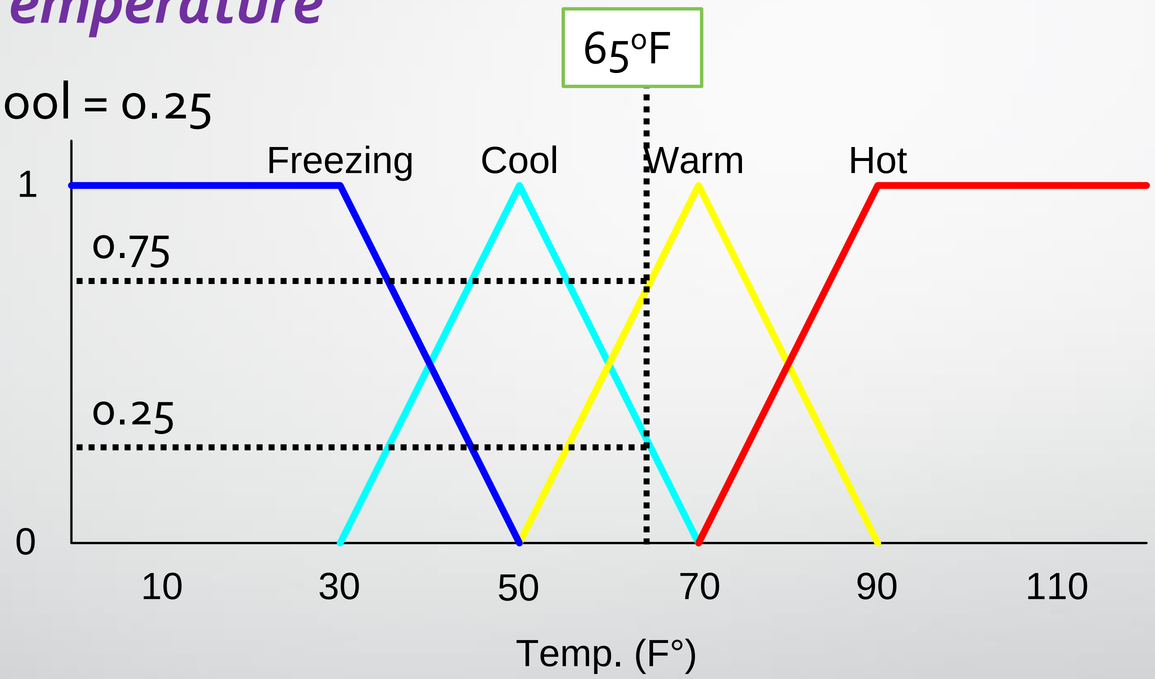


Fuzzy Systems Design : Example

- **Q: How fast can I drive today?**
- **Let's find the answer for:**
  - What is the system decision for my **speed** when the **temperature** is **65°F** and it is **25% cloud cover**?

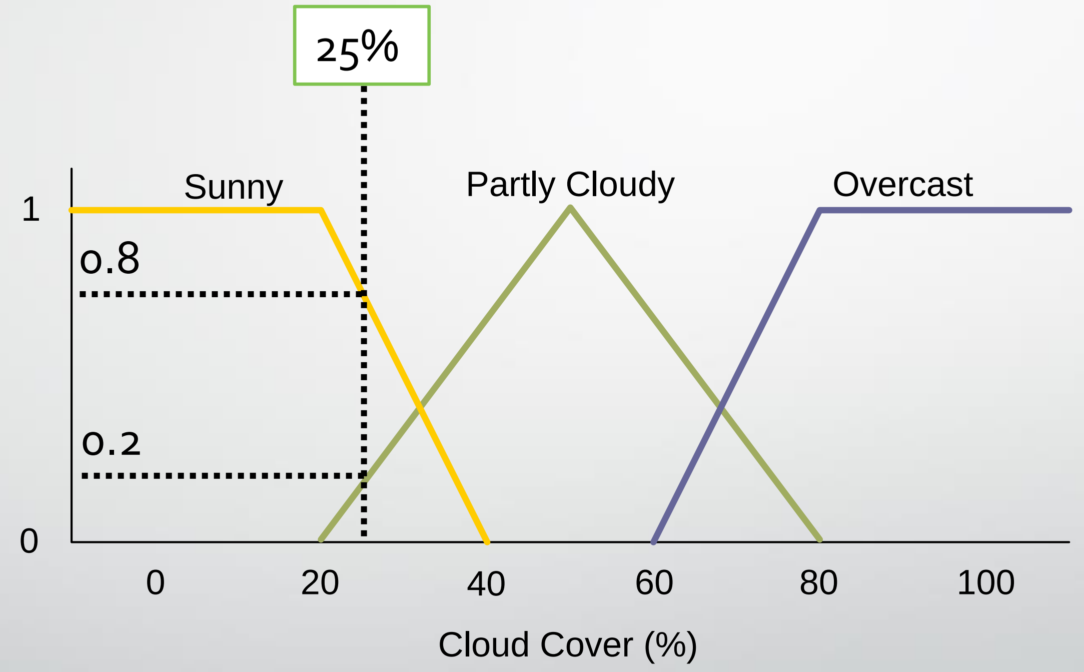
Fuzzy Systems Design : Example

- **Fuzzification**: ***Temperature***
  - Warm= 0.75, Cool = 0.25



Fuzzy Systems Design : Example

- **Fuzzification: *Cloud Cover***
  - Sunny = 0.8 and Cloudy = 0.2



Fuzzy Systems Design

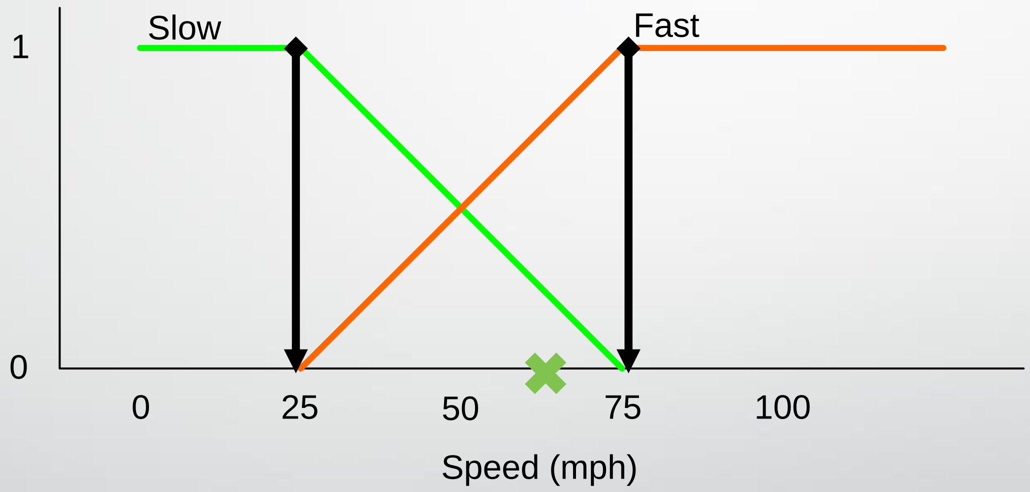
- **Rule 1**: **Sunny(Cover)$\wedge$Warm(Temp) $\Rightarrow$ Fast(Speed)**
  - Sunny(Cover) = 0.8
  - Warm(Temp) = 0.75
  - Fast(Speed) = 0.8$\wedge$0.75 = 0.75

Fuzzy Systems Design

- **Rule 2: Cloudy(Cover)$\wedge$Cool(Temp) $\Rightarrow$ Slow(Speed)**
  - Cloudy(Cover)= 0.2
  - Cool(Temp)= 0.25
  - Slow(Speed) = 0.2$\wedge$0.25 = 0.2

Fuzzy Systems Design : Example

- **Defuzzification**: The outputs of all of the fuzzy rules must now be combined to obtain one fuzzy output distribution and finds its **centre of mass** to come up with ***one crisp number***



- **Speed** $= \dfrac{0.2\text{xSlow} + 0.75\text{xFast}}{0.2+0.75} = \dfrac{0.2\text{x}25 + 0.75\text{x}75}{0.95}$ = 64.4 mph

Fuzzy Systems Design : Example

- **It's your turn!!** Same question, How fast can I drive today?
- Determine the **speed** when it is **70°F** and **50% cloudy**?

Weaknesses ☹

- Requires tuning of membership functions
- May not scale well to large or complex problems
- Deals with imprecision, and vagueness, but not uncertainty

Class Activity – Survey (15 min) – COMP8811

Questions