INTRODUCTION TO ARTIFICIAL INTELLIGENCE

Objectives of this Course

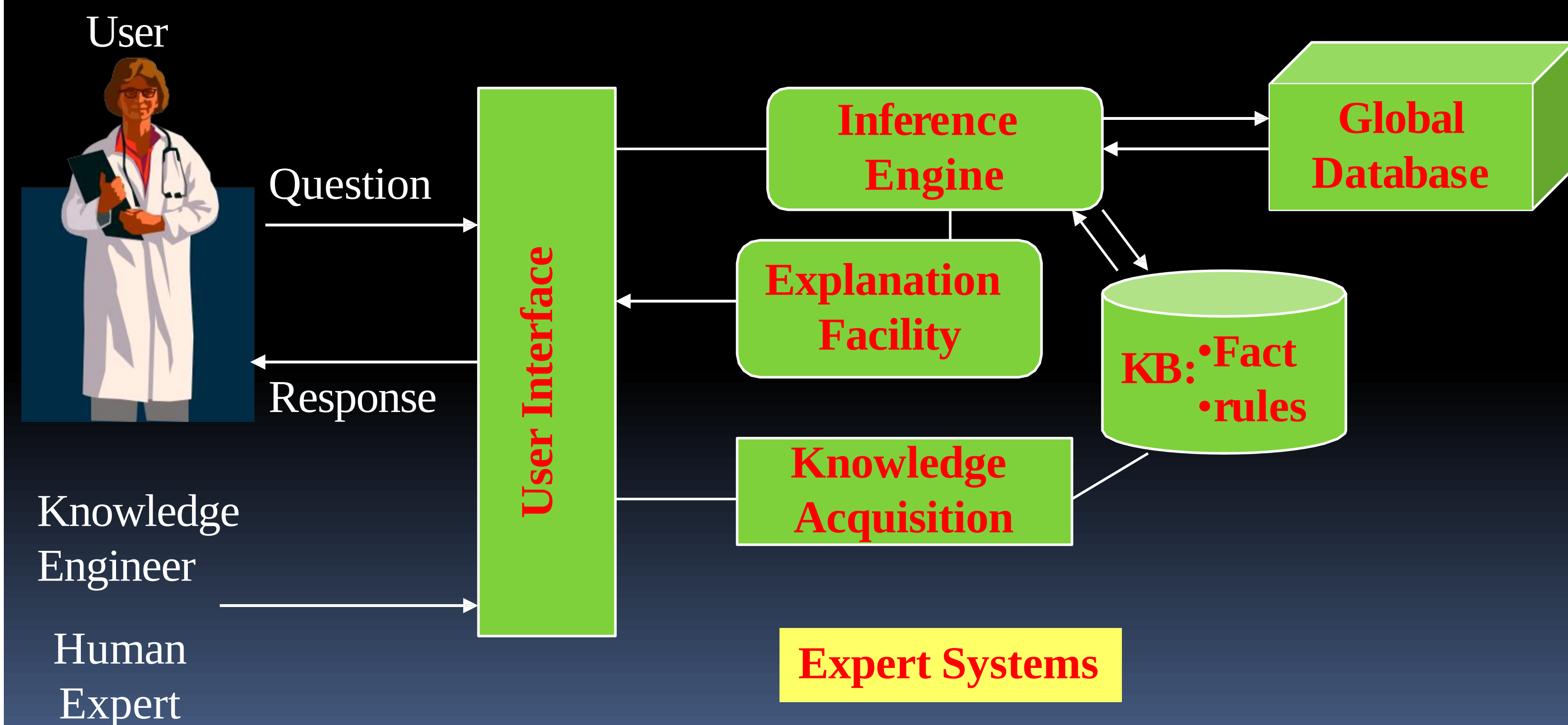
- This class is a broad introduction to artificial intelligence (AI)
  - AI is a very broad field with many subareas
    - We will cover many of the primary concepts/ideas
    - But in 15 weeks we can't cover everything

Today's Lecture

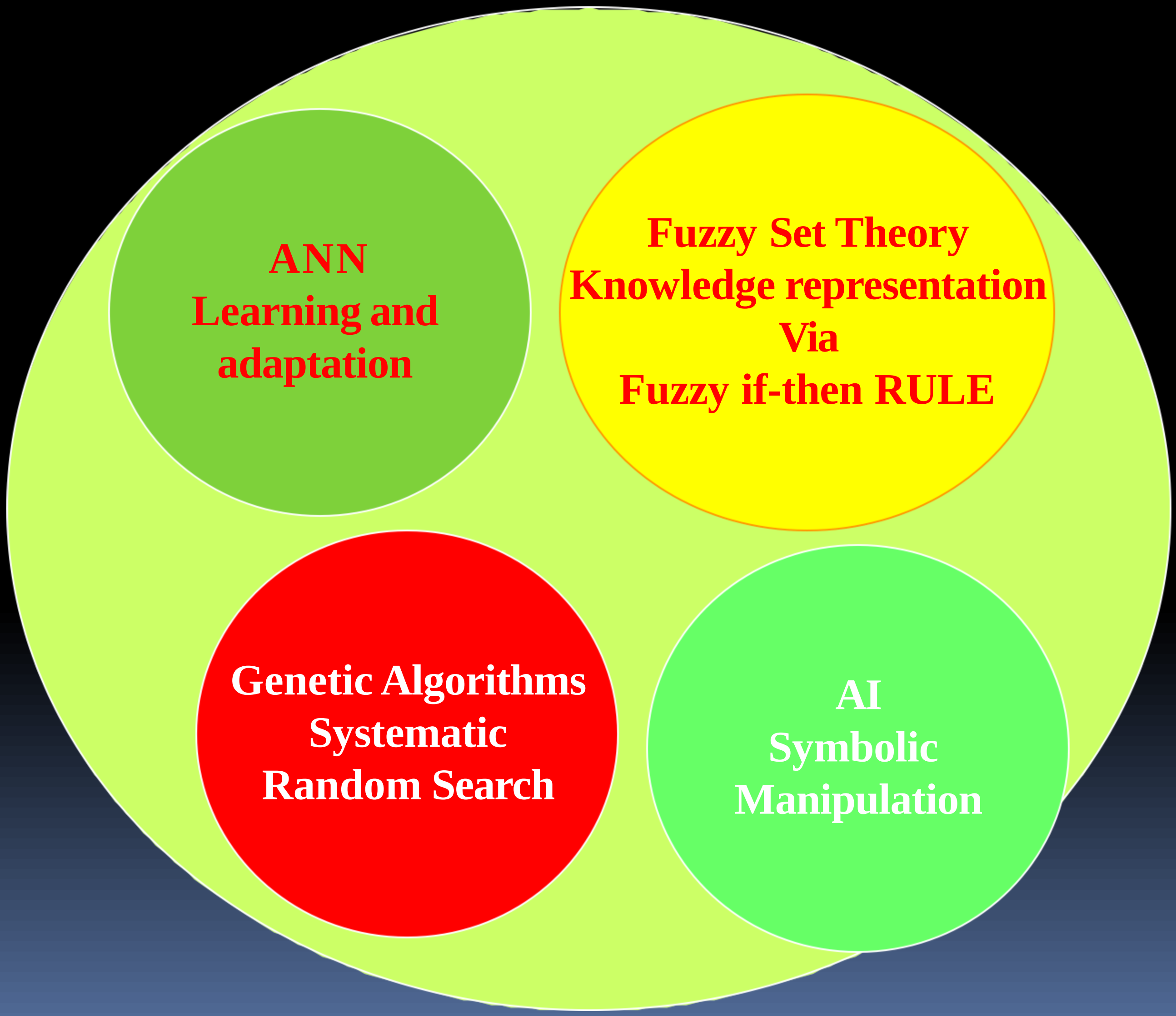
- What is intelligence? What is artificial intelligence?
- A very brief history of AI
  - Modern successes: Stanley the driving robot
- An AI scorecard
  - How much progress has been made in different aspects of AI
- AI in practice
  - Successful applications

AI and Soft Computing: A Different Perspective

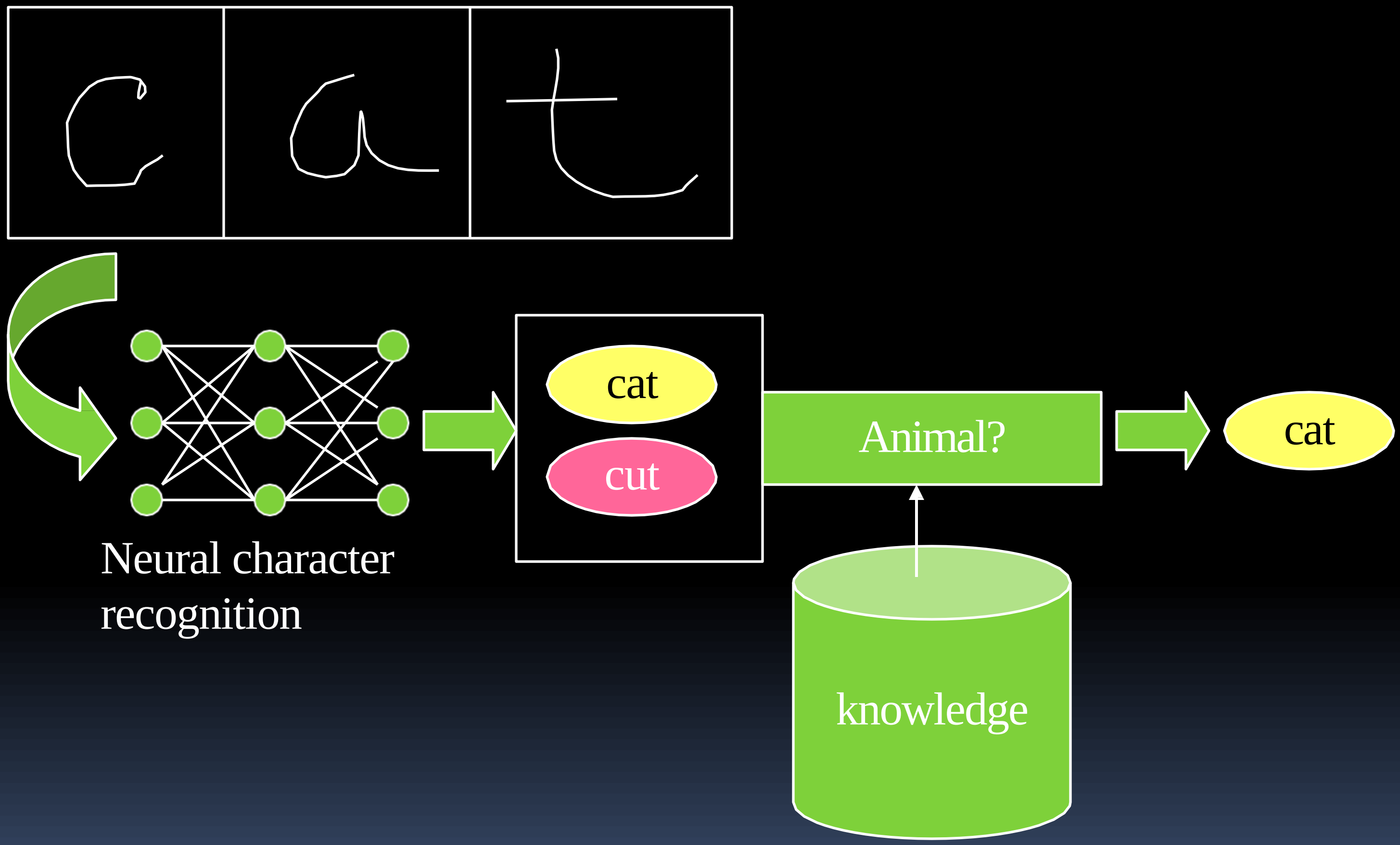
- AI: predicate logic and symbol manipulation techniques



AI and Soft Computing



AI and Soft Computing



What is Hard Computing ?

- **Hard computing, i.e., conventional computing, requires a precisely stated analytical model and often a lot of Computational Time.**

- **Many analytical models are valid for ideal cases.**

- **Real world problems exist in a non-ideal environment.**

Premises and guiding principles of Hard Computing

Precision, Certainty, and Rigor.

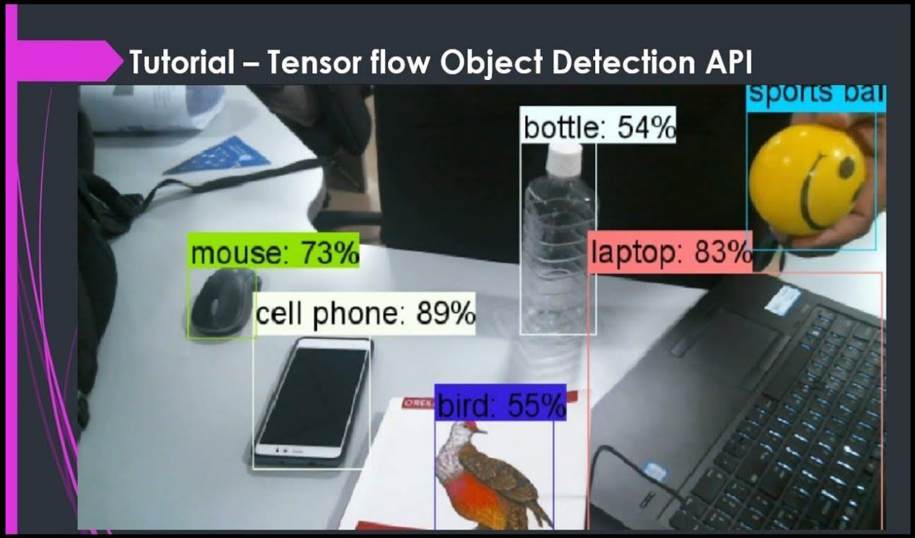
- **Many contemporary problems do not lend themselves to precise solutions such as:**

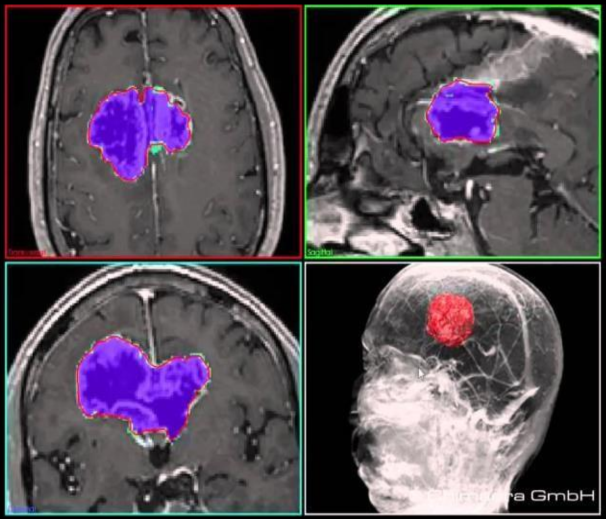
Recognition problems (handwriting, speech, objects, images, texts)

Mobile robot coordination, forecasting, combinatorial problems etc.

Reasoning on natural languages

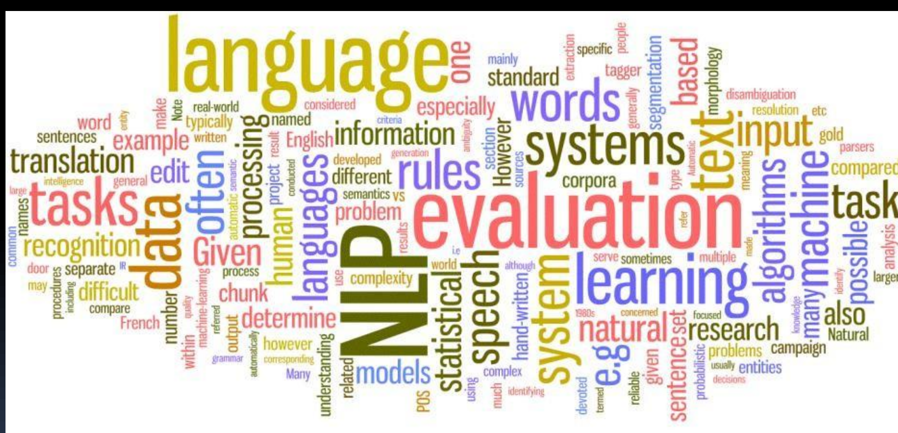
Recognition problems (handwriting, speech, objects, images, texts)





Mobile robot Coordination

weather forecasting



Natural Language Processing

What is Artificial Intelligence?

Some Definitions (I)

The exciting new effort to make computers think … *machines with minds*, in the full literal sense.

Haugeland, 1985

Some Definitions (II)

The study of mental faculties through the use of computational models.

Charniak and McDermott, 1985

A field of study that seeks to explain and emulate intelligent behavior in terms of computational processes.

Schalkoff, 1990

Some Definitions (III)

The study of how to make computers do things at which, at the moment, people are better.

Rich & Knight, 1991

Outline of the Course

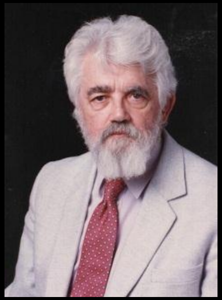
- Knowledge representation:
  - propositional logic and first-order logic
  - inference in Expert Systems
  - Fuzzy logic
  - Rough set
  - Machine learning: classification trees
  - Neural networks
  - Ohers ?

What is ntelligence?

- Intelligence:
  - "the capacity to learn and solve problems" (Websters dictionary)
  - in particular,
    - *the ability to solve novel problems*
    - *the ability to act rationally*
    - *the ability to act like humans*

- Artificial Intelligence
  - build and understand intelligent entities or agents
  - 2 main approaches: "engineering" versus "cognitive modeling"

What is Artificial Intelligence?



(John McCarthy, Stanford University)

- **What is artificial intelligence?**

  It is the science and engineering of making intelligent machines, especially intelligent computer programs. It is related to the similar task of using computers to understand human intelligence, but AI does not have to confine itself to methods that are biologically observable.

- **Isn't there a solid definition of intelligence that doesn't depend on relating it to human intelligence?**

  Not yet. The problem is that we cannot yet characterize in general what kinds of computational procedures we want to call intelligent. We understand some of the mechanisms of intelligence and not others.

- More in: http://www-formal.stanford.edu/jmc/whatisai/node1.html

What’s involved in Intelligence?

- Ability to interact with the real world
  - to perceive, understand, and act
  - e.g., speech recognition and understanding and synthesis
  - e.g., image understanding
  - e.g., ability to take actions, have an effect

- Reasoning and Planning
  - modeling the external world, given input
  - solving new problems, planning, and making decisions
  - ability to deal with unexpected problems, uncertainties

- Learning and Adaptation
  - we are continuously learning and adapting
  - our internal models are always being “updated”
    - e.g., a baby learning to categorize and recognize animals

Academic Disciplines important to AI.

- Mathematics — Formal representation and proof, algorithms, computation, (un)decidability, (in)tractability, probability.
- Economics — utility, decision theory, rational economic agents
- Neuroscience — neurons as information processing units.
- Psychology/ Cognitive Science — how do people behave, perceive, process information, represent knowledge.
- Computer engineering — building fast computers
- Control theory — design systems that maximize an objective function over time
- Linguistics — knowledge representation, grammar

History of AI

- 1943: early beginnings
  - McCulloch & Pitts: Boolean circuit model of brain
- 1950: Turing
  - Turing's "Computing Machinery and Intelligence“
- 1956: birth of AI
  - Dartmouth meeting: "Artificial Intelligence“ name adopted
- 1950s: initial promise
  - Early AI programs, including
  - Samuel's checkers program
  - Newell & Simon's Logic Theorist

History of AI

- 1966—73: Reality dawns
  - Realization that many AI problems are intractable
  - Limitations of existing neural network methods identified
    - Neural network research almost disappears
- 1969—85: Adding domain knowledge
  - Development of knowledge-based systems
  - Success of rule-based expert systems,
    - E.g., DENDRAL, MYCIN
    - But were brittle and did not scale well in practice
- 1986-- Rise of machine learning
  - Neural networks return to popularity
  - Major advances in machine learning algorithms and applications
- 1990-- Role of uncertainty
  - Bayesian networks as a knowledge representation framework
- 1995-- AI as Science
  - Integration of learning, reasoning, knowledge representation
  - AI methods used in vision, language, data mining, etc

Different Types of Artificial Intelligence

1. Modeling exactly how humans actually think
2. Modeling exactly how humans actually act
3. Modeling how ideal agents “should think”
4. Modeling how ideal agents “should act”

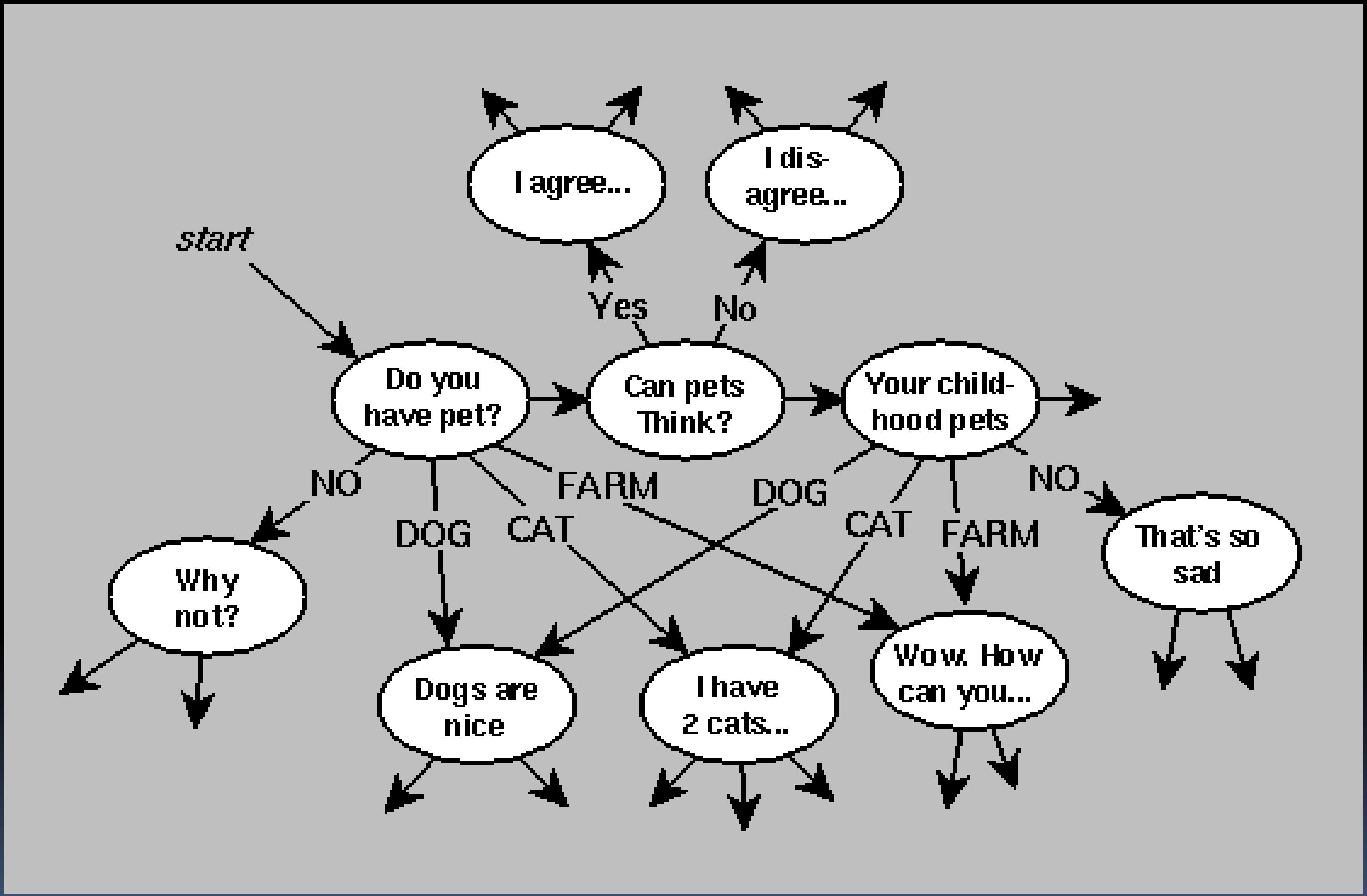
- Modern AI focuses on the last definition
  - we will also focus on this “engineering” approach
  - success is judged by how well the agent performs

The Origins of AI

- 1950 Alan Turing's paper, *Computing Machinery and Intelligence*, described what is now called "The Turing Test".

- Turing predicted that in about fifty years "an average interrogator will not have more than a 70 percent chance of making the right identification after five minutes of questioning".

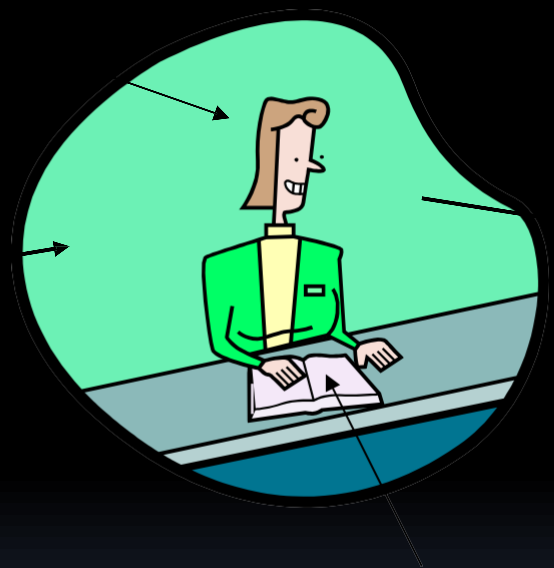
- 1957 Newell and Simon predicted that "Within ten years a computer will be the world's chess champion."


start
Do you have pet?
Can pets Think?
Your child-hood pets
Yes
No
I agree...
I dis-agree...
NO
DOG
CAT
FARM
Why not?
Dogs are nice
I have 2 cats...
Wow. How can you...
That's so sad

The Chinese Room

She does not know Chinese

Chinese Writing is given to the person



Correct Responses

Set of rules, in English, for transforming phrases

The Chinese Room

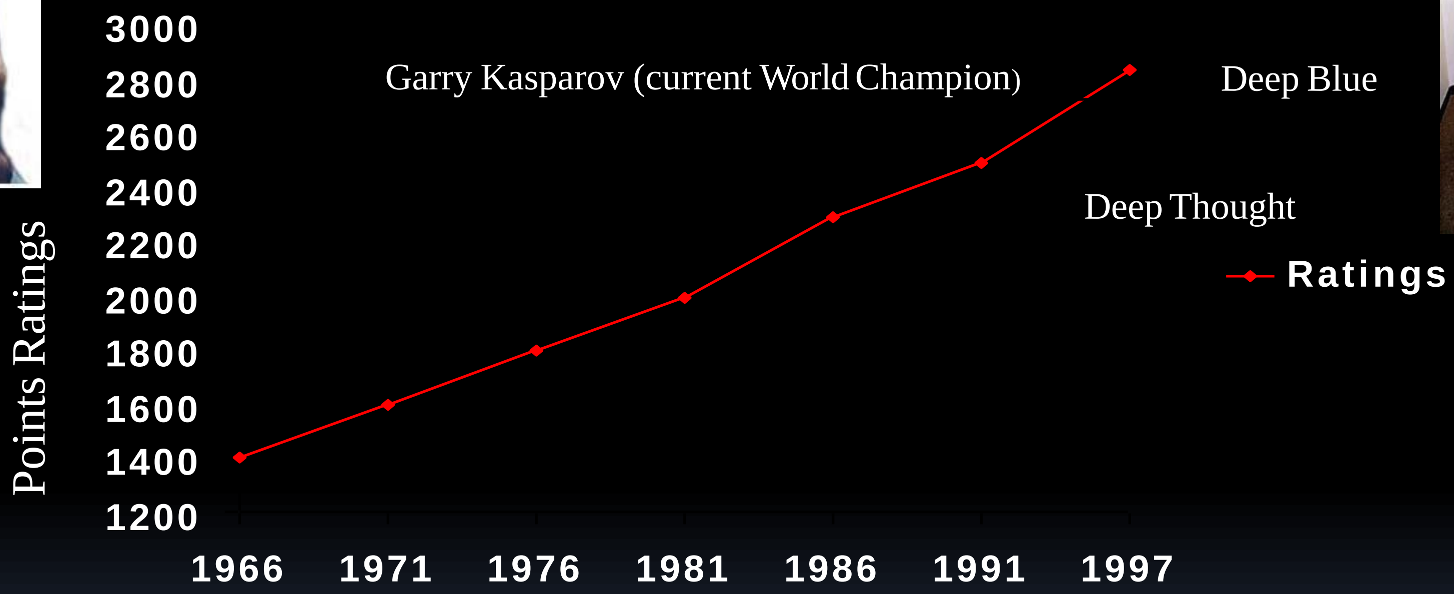
- So imagine an individual is locked in a room and given a batchof Chinese writing.
- The person locked in the room does not understand Chinese. Next he is given more Chinese writing and a set of rules (in English which he understands) on how to collate the first set of Chinese characters with the second set of Chinese characters.
- Suppose the person gets so good at manipulating the Chinese symbols and the rules are so good, that to those outside the room it appears that the person understands Chinese.
- Searle's point isthat, he doesn't really understand Chinese, it really only following a set of rules.
- Following this argument, a computer could never be truly intelligent, it is only manipulating symbols that it really doesn't understand the semanticcontext.

Can these Questions are Answerable?

- Can Computers play Humans at Chess?
- Can Computers Talk?
- Can Computers Recognize Speech?
- Can Computers Learn and Adapt ?
- Can Computers “see”?
- Can Computers plan and make decisions?

Can Computers play Humans at Chess?

- Chess Playing is a classic AI problem
  - well-defined problem
  - very complex: difficult for humans to play well



- Conclusion: YES: today’s computers can beat even the best human

Can Computers Talk?

- This is known as "speech synthesis"
  - translate text to phonetic form
    - e.g., "fictitious" -> fik-tish-es
  - use pronunciation rules to map phonemes to actual sound
    - e.g., "tish" -> sequence of basic audio sounds
- Difficulties
  - sounds made by this "lookup" approach sound unnatural
  - sounds are not independent
    - e.g., "act" and "action"
    - modern systems (e.g., at AT&T) can handle this pretty well
  - a harder problem is emphasis, emotion, etc
    - humans understand what they are saying
    - machines don't: so they sound unnatural
- Conclusion:
  - NO, for complete sentences
  - YES, for individual words

Can Computers Recognize Speech?

- Speech Recognition:
  - mapping sounds from a microphone into a list of words
  - classic problem in AI, very difficult

- Recognizing single words from a small vocabulary
  - systems can do this with high accuracy (order of 99%)
  - e.g., directory inquiries
    - limited vocabulary (area codes, city names)
    - computer tries to recognize you first, if unsuccessful hands you over to a human operator
    - saves millions of dollars a year for the phone companies

Recognizing human speech (ctd.)

- Recognizing normal speech is much more difficult
  - speech is continuous: where are the boundaries between words?
    - e.g., “John’s car has a flat tire”
  - large vocabularies
    - can be many thousands of possible words
    - we can use **context** to help figure out what someone said
      - e.g., hypothesize and test
      - try telling a waiter in a restaurant:
        “I would like some sugar in my coffee”
  - background noise, other speakers, accents, colds, etc
  - on normal speech, modern systems are only about 60-70% accurate

- Conclusion:
  - NO, normal speech is too complex to accurately recognize
  - YES, for restricted problems (small vocabulary, single speaker)

Can Computers Learn and Adapt ?

- Learning and Adaptation
  - consider a computer learning to drive on the freeway
  - we could code lots of rules about what to do
  - and/or we could have it learn from experience

  - **machine learning** allows computers to learn to do things without explicit programming

- Conclusion: YES, computers can learn and adapt, when presented with information in the appropriate way

Can Computers "see"?

- Recognition v. Understanding (like Speech)
  - Recognition and Understanding of Objects in a scene
    - look around this room
    - you can effortlessly recognize objects
    - human brain can map 2d visual image to 3d "map"
- Why is visual recognition a hard problem?
- Conclusion:
  - mostly NO: computers can only "see" certain types of objects under limited circumstances
  - YES for certain constrained problems (e.g., face recognition)

Can Computers plan and make decisions?

- Intelligence
  - involves solving problems and making decisions and plans
  - e.g., you want to visit your cousin in Boston
    - you need to decide on dates, flights
    - you need to get to the airport, etc
    - involves a sequence of decisions, plans, and actions
- What makes planning hard?
  - the world is not predictable:
    - your flight is canceled or there's a backup on the 405
  - there is a potentially huge number of details
    - do you consider all flights? all dates?
      - no: commonsense constrains your solutions
  - AI systems are only successful in constrained planning problems
- Conclusion: NO, real-world planning and decision-making is still beyond the capabilities of modern computers
  - exception: very well-defined, constrained problems: mission planning for satelites.

Summary of State of AI Systems in Practice

- Speech synthesis, recognition and understanding
  - very useful for limited vocabulary applications
  - unconstrained speech understanding is still too hard
- Computer vision
  - works for constrained problems (hand-written zip-codes)
  - understanding real-world, natural scenes is still too hard
- Learning
  - adaptive systems are used in many applications: have their limits
- Planning and Reasoning
  - only works for constrained problems: e.g., chess
  - real-world is too complex for general systems
- Overall:
  - many components of intelligent systems are "doable"
  - there are many interesting research problems remaining

Intelligent Systems in Your Everyday Life

- Post Office
  - automatic address recognition and sorting of mail
- Banks
  - automatic check readers, signature verification systems
  - automated loan application classification
- Telephone Companies
  - automatic voice recognition for directory inquiries
- Credit Card Companies
  - automated fraud detection
- Computer Companies
  - automated diagnosis for help-desk applications
- Netflix:
  - movie recommendation
- Google:
  - Search Technology

AI Applications: Identification Technologies

- ID cards
  - e.g., ATM cards
  - can be a nuisance and security risk:
    - cards can be lost, stolen, passwords forgotten, etc
- Biometric Identification
  - walk up to a locked door
    - camera
    - fingerprint device
    - microphone
    - iris scan
  - computer uses your biometric signature foridentification
    - face, eyes, fingerprints, voice pattern, iris pattern

The agenda of AI class:

1. Fuzzy logic
2. Prepositional logic – prolog – expert systems with inference algorithms
3. Rough set theory
4. Decision trees, kNN, Naive Bayes
5. Neural network

Problem Solving, Search and Control Strategies

Problem Solving, Search and Control Strategies

1. **General Problem Solving**
   - Problem solving definitions:
   - problem space,
   - problem solving,
   - state space,
   - state change,
   - structure of state space,
   - problem solution,
   - problem description;
   - Examples of problem definition.

Problem Solving, Search and Control Strategies

- **Problem definitions:**

A *problem* is defined by its *elements* and their *relations*.

To provide a formal description of a problem, we need to do following:

a. Define a *state space* that contains all the possible configurations of the relevant objects, including some impossible ones.
b. Specify one or more states, that describe possible situations, from which the problem-solving process may start. These states are called *initial states*.
c. Specify one or more states that would be acceptable solution to the problem. These states are called *goal states*.
d. Specify a set of *rules* that describe the *actions* (*operators*) available.

The problem can then be solved by using the *rules*, in combination with an appropriate *control strategy*, to move through the *problem space* until a *path* from an *initial state* to a *goal state* is found.

Problem Solving, Search and Control Strategies

- Problem definitions:

  This process is known as ***search***.

  - Search is fundamental to the problem-solving process.
  - Search is a general mechanism that can be used when more direct method is not known.
  - Search provides the framework into which more direct methods for solving subparts of a problem can be embedded.

  A very large number of AI problems are formulated as search problems.

Problem Solving, Search and Control Strategies

Problem Space

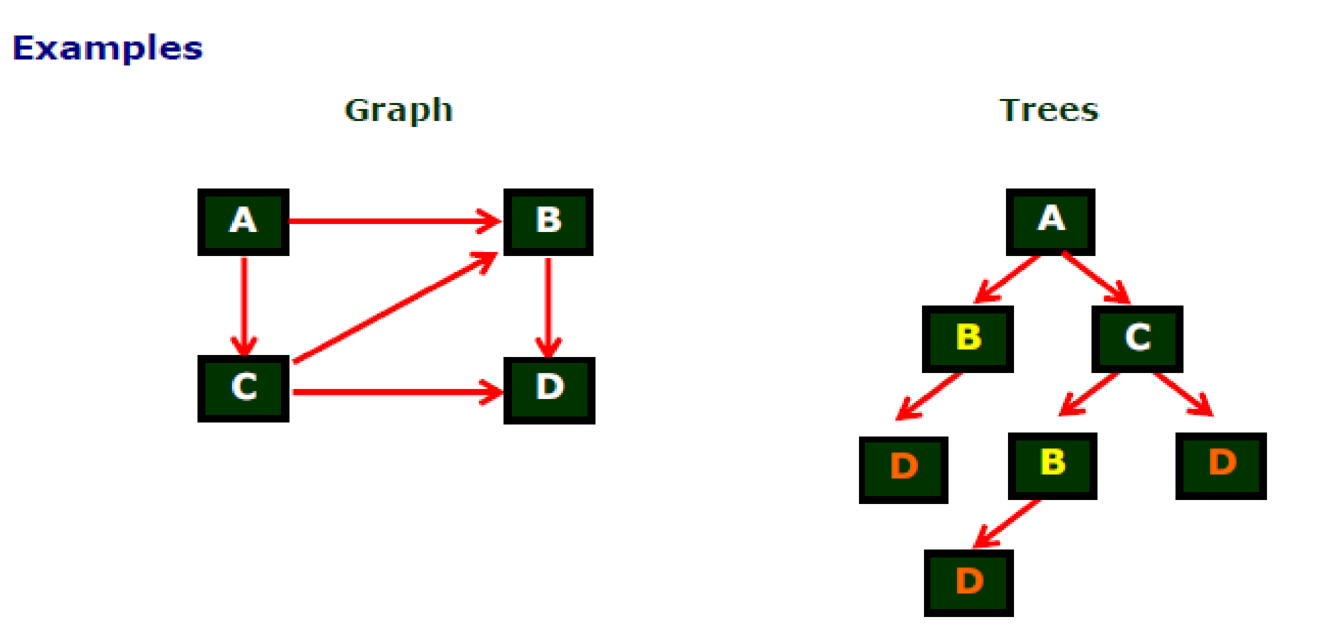
- A *problem space* is represented by directed *graph*, where *nodes* represent *search state* and *paths* represent the *operators* applied to change the *state*.

- To simplify a search algorithms, it is often convenient to logically and programmatically represent a problem space as a ***tree***.
- A tree usually decreases the complexity of a search at a ***cost***. Here, the cost is due to duplicating some nodes on the tree that were linked numerous times in the graph; e.g., node **B** and node **D** shown in example below.

Problem Solving, Search and Control Strategies

Problem Space

A **tree** is a graph in which any two vertices are connected by exactly one path. Alternatively, any connected graph with no cycles is a tree.



Problem Solving, Search and Control Strategies

- ***States***

A state is a representation of elements at a given moment. A problem is defined by its elements and their relations.
At each instant of a problem, the elements have specific descriptors and relations; the descriptors tell - how to select elements ?
Among all possible states, there are two special states called :

- *Initial state* *is the start point*
- *Final state* *is the goal state*

Problem Solving, Search and Control Strategies

- **State Change:** Successor Function

A *Successor Function* is needed for state change.

The successor function moves one state to another state.

Successor Function :

- ◇ Is a description of possible actions; a set of operators.
- ◇ Is a transformation function on a state representation, which converts that state into another state.
- ◇ Defines a relation of accessibility among states.
- ◇ Represents the conditions of applicability of a state and corresponding transformation function

Problem Solving, Search and Control Strategies

- **State Space**

A *State space* is the set of all states reachable from the

initial state. Definitions of terms :

◇ A *state space* forms a *graph* (or map) in which the *nodes* are states and the *arcs* between nodes are actions.

◇ In *state space*, a *path* is a sequence of states connected by a sequence of actions.

◇ The *solution* of a problem is part of the map formed by the *state space*.

Problem Solving, Search and Control Strategies

- ## Structure of a State Space

The *Structures* of *state space* are *trees* and *graphs*.

- Tree is a hierarchical structure in a graphical form; and
- Graph is a non-hierarchical structure.

◇ **Tree** has only one path to a given node;
i.e., a *tree* has one and only one path from any point to any other point.

◇ **Graph** consists of a set of nodes (vertices) and a set of edges (arcs).
Arcs establish relationships (connections) between the nodes; i.e., a graph has several paths to a given node.

◇ **operators** are directed *arcs* between nodes.

Search process explores the *state space*. In the worst case, the search explores all possible *paths* between the *initial state* and the *goal state*.

Problem Solving, Search and Control Strategies

- **Problem Solution**

  In the *state space*, a solution is a path from the *initial state* to a *goal state* or sometime just a *goal state*.

  ◇ A Solution cost function assigns a numeric cost to each path; It also gives the cost of applying the operators to the states.

  ◇ A Solution quality is measured by the path cost function; and An optimal solution has the lowest path cost among all solutions.

  ◇ The solution may be any or optimal or all.

  ◇ The importance of cost depends on the problem and the type of solution asked.

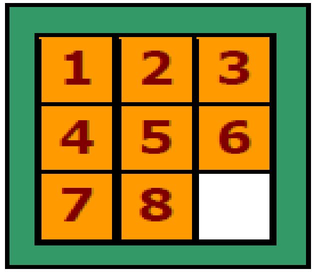
Problem Solving, Search and Control Strategies

1. Examples of Problem Definitions

- **Example 1 :**

  A game of 8–Puzzle

  - State space : configuration of **8 - tiles** on the board
  - Initial state : any configuration
  - Goal state : tiles in a specific order
  - Action : "blank moves"
    - Condition: the move is within the board
    - Transformation: blank moves Left, Right, Up, Dn
  - Solution : optimal sequence of operators



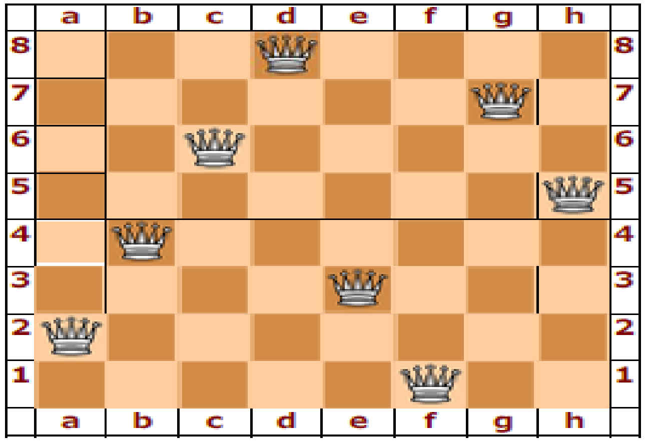
Solution

Problem Solving, Search and Control Strategies

- **Example 2 :**

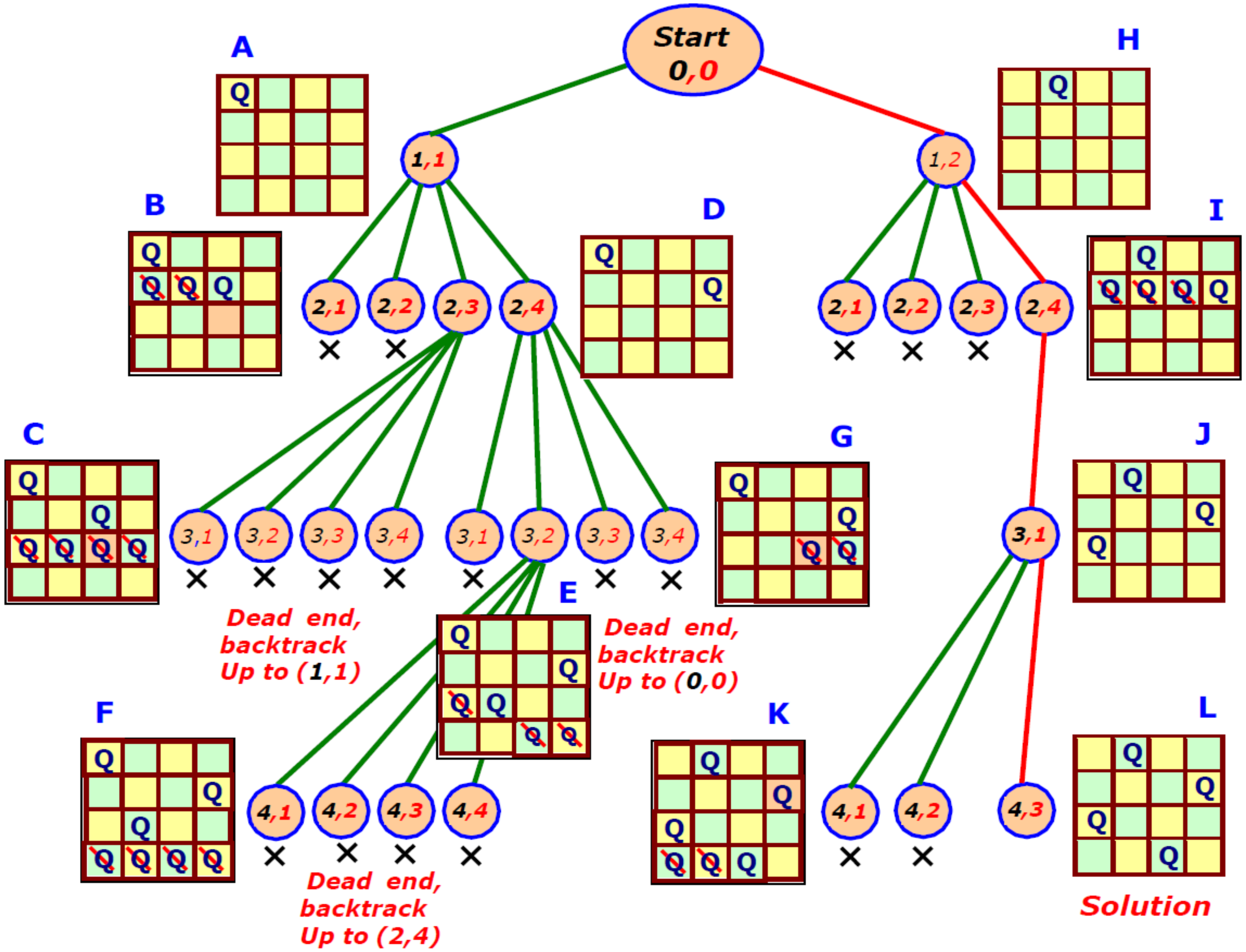
A game of n - queens puzzle; n = 8

- State space : configurations **n = 8** queens on the board with only one queen per row and column
- Initial state : configuration without queens on the board
- Goal state : configuration with **n = 8** queens such that no queen attacks any other
- Operators or actions : place a queen on the board.
  - Condition: the new queen is not attacked by any other already placed
  - Transformation: place a new queen in a particular square of the board
- Solution : one solution (cost is not considered)



One Solution

Example : Backtracking to solve **N = 4** Queens problem.



Hierarchical Representation of Search Algorithms

A representation of most search algorithms is illustrated below. It begins with two types of search - Uninformed and Informed.

Uninformed Search : Also called *blind, exhaustive or brute-force* search, uses no information about the problem to guide the search and therefore may not be very efficient.

Informed Search : Also called *heuristic* or *intelligent* search, uses information about the problem to guide the search, usually guesses the distance to a goal state and therefore efficient, but the search may not be always possible.

always possible.

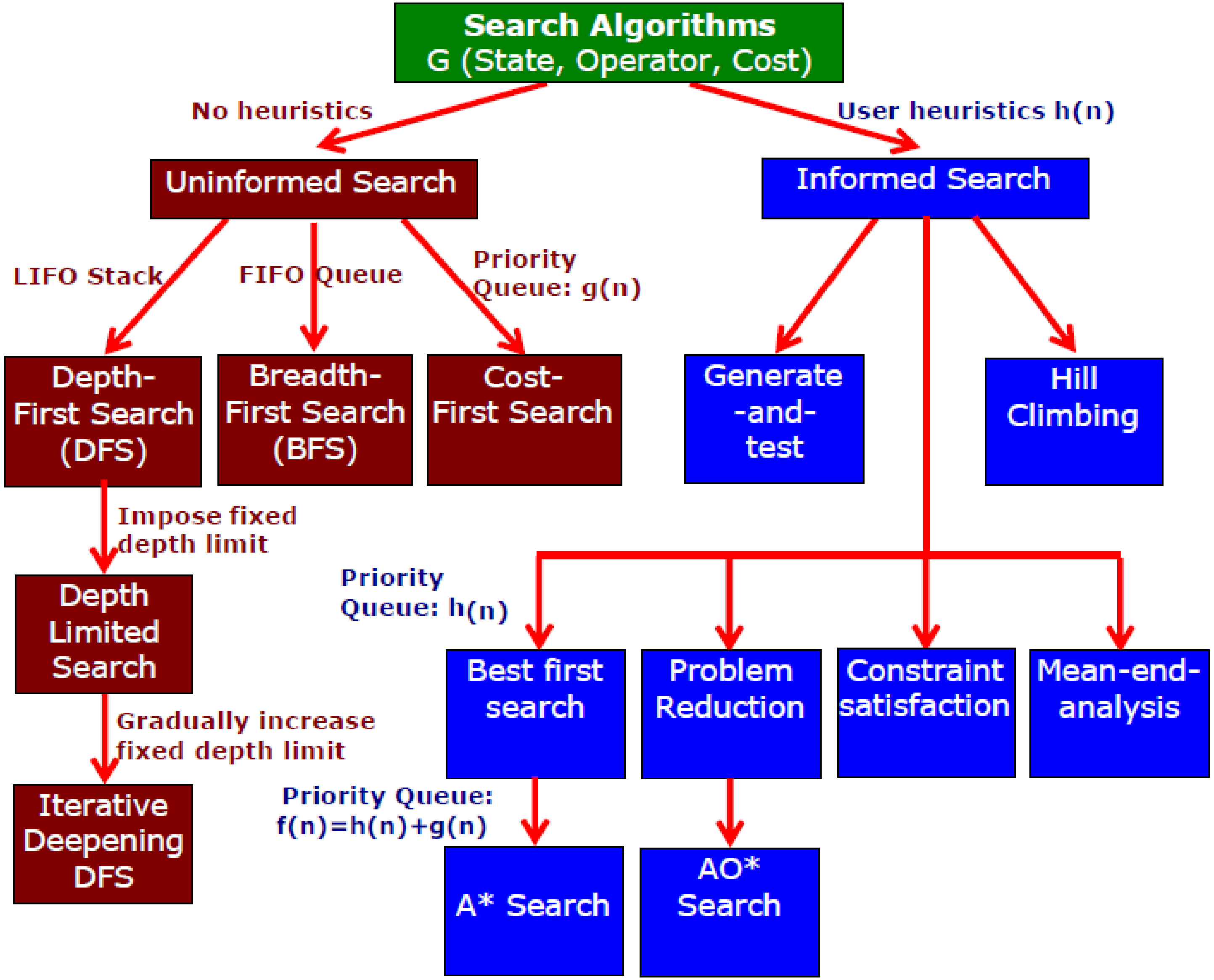


Fig. Different Search Algorithms

Depth-First Search (DFS)

Here explained the Depth-first search tree, the backtracking to the previous level, and the Depth-first search algorithm

- DFS explores a path all the way to a leaf before backtracking and exploring another path.
- **Example:** Depth-first search tree

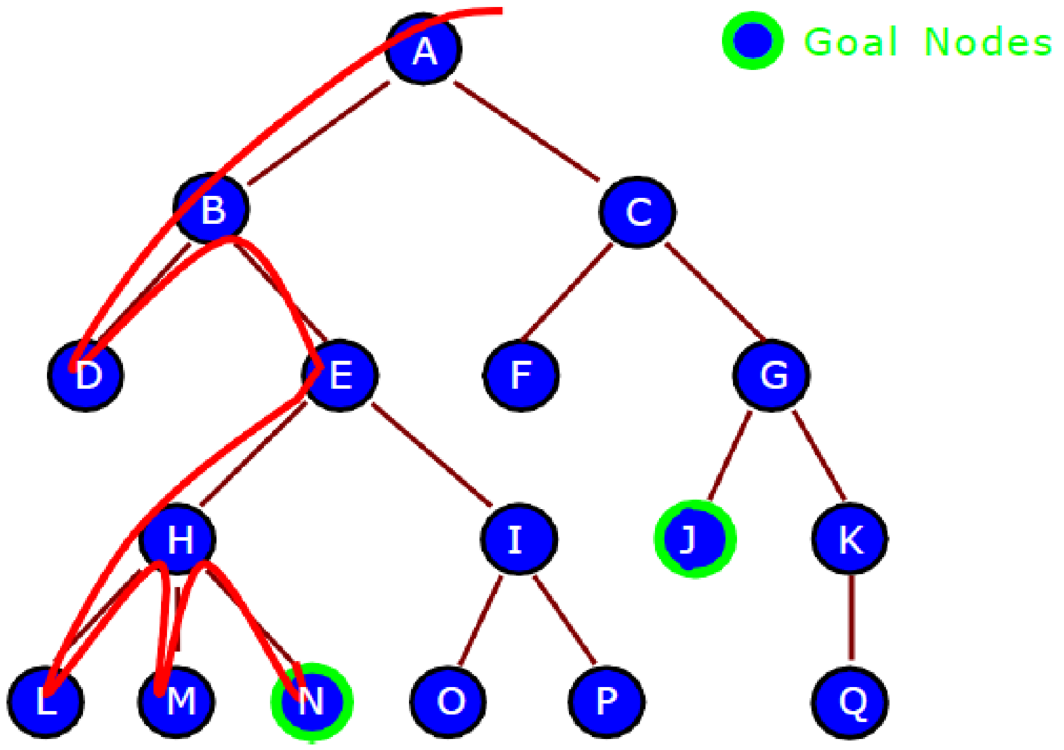


Fig. Depth-first search (DFS)

A B D E H L M N I O P C F G J K Q

- After searching node **A**, then **B**, then **D**, the search *backtracks* and tries another path from node **B**.

- The goal node **N** will be found before the goal node **J**.

◇ **Algorithm** – Depth-first search

- Put the root node on a stack;
  while (stack is not empty)
  { remove a node from the stack;
  if (node is a goal node) return success;
  put all children of node onto the stack; }
  return failure;

Note :

‡ At every step, the stack contains some nodes from each level.

‡ The stack size required depends on the branching factor **b**.

‡ Searching level **n**, the stack contains approximately **b** * **n** nodes.

‡ When this method succeeds, it does not give the path.

‡ To hold the search path the algorithm required is "*Recursive depth-first search*" and stack size large.

Fuzzy Set Theory

UNIT-2

Introduction

- The word “fuzzy” means “vaguness (ambiguity)”.
- Fuzziness occurs when the boundary of a piece of information is not clear-cut.
- Fuzzy sets - 1965 Lotfi Zadeh as an extension of classical notation set.
- Classical set theory allows the membership of the elements in the set in **binary terms**.
- Fuzzy set theory permits membership function valued in the interval [0,1].

Introduction

Example:

Words like young, tall, good or high are fuzzy.

- There is no single quantitative value which defines the term young.
- For some people, age 25 is young, and for others, age 35 is young.
- The concept young has no clean boundary.
- Age 35 has some possibility of being young and usually depends on the context in which it is being considered.

Fuzzy set theory is an extension of classical set theory where elements have degree of membership.

Introduction

- In real world, there exist much fuzzy knowledge (i.e. vague, uncertain inexact etc).
- Human **thinking** and **reasoning** (analysis, logic, interpretation) frequently involved **fuzzy** information.
- Human can give satisfactory answers, which are probably true.
- Our systems are unable to answer many question because the systems are designed based upon classical set theory (Unreliable and incomplete).
- We want, our system should be able to cope with unreliable and incomplete information.
- Fuzzy system have been provide solution.

Introduction

Classical set theory

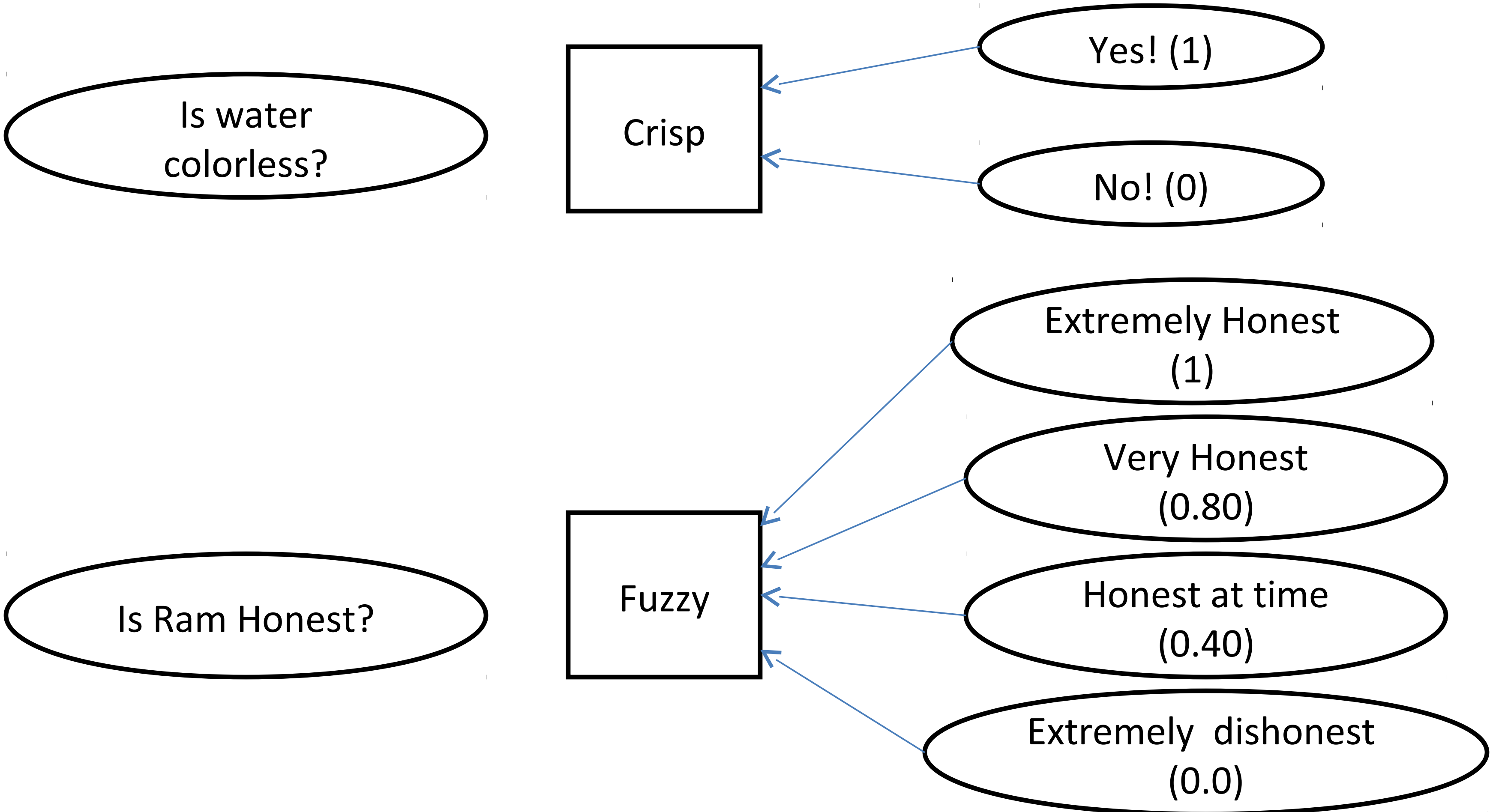
- Classes of objects with sharp boundaries.
- A classical set is defined by crisp(exact) boundaries, i.e., there is no uncertainty about the location of the set boundaries.
- Widely used in digital system design

Fuzzy set theory

- Classes of objects with un-sharp boundaries.
- A fuzzy set is defined by its ambiguous boundaries, i.e., there exists uncertainty about the location of the set boundaries.
- Used in fuzzy controllers.

Introduction (Continue)

Example



Fuzzy vs crips

Classical set theory

- A Set is any well defined collection of objects.
- An object in a set is called an element or member of that set.
- Sets are defined by a simple statement,
- Describing whether a particular element having a certain property belongs to that particular set.

$$A = \{a_1, a_2, a_3, \ldots\ldots, a_n\}$$

- If the elements a_i ($i = 1,2,3,\ldots.,n$) of a set A are subset of universal set X, then set A can be represented for all elements $x \in X$ by its characteristics function

$$\mu_A(x) = 1 \text{ if } x \in X \text{ otherwise } 0$$

Operations on classical set theory

Union: the union of two sets A and B is given as

$$A \cup B = \{ x \mid x \in A \text{ or } x \in B \}$$

Intersection: the intersection of two sets A and B is given as

$$A \cap B = \{ x \mid x \in A \text{ and } x \in B \}$$

Complement: It is denoted by Ã and is defined as

$$\tilde{A} = \{ x \mid x \text{ does not belongs } A \text{ and } x \in X \}$$

Fuzzy Sets

- Fuzzy sets theory is an extension of classical set theory.
- Elements have varying degree of membership. A logic based on two truth values,
- ***True*** and ***False*** is sometimes insufficient when describing human reasoning.
- Fuzzy Logic uses the whole interval between 0 (false) and 1 (true) to describe human reasoning.
- A Fuzzy Set is any set that allows its members to have different degree of membership, called **membership function**, having interval [0,1].

Fuzzy Sets

- **Fuzzy Logic** is derived from fuzzy set theory
- Many degree of membership (between 0 to 1) are allowed.
- Thus a membership function $\boldsymbol{\mu}_A^{(x)}$ is associated with a fuzzy sets Ã such that the function maps every element of universe of discourse X to the interval [0,1].
- The mapping is written as: $\mu_{\tilde{A}}(x): X \rightarrow [0,1]$.

- Fuzzy Logic is capable of handing inherently imprecise (vague or inexact or rough or inaccurate) concepts

Fuzzy Sets

- **Fuzzy set** is defined as follows:

- If X is an universe of discourse and x is a particular element of X, then a fuzzy set A defined on X and can be written as a collection of ordered pairs

$$A = \{(x, \mu_{\tilde{A}}(x)), x \in X \}$$

Fuzzy Sets (Continue)

Example

- Let X = {g_1, g_2, g_3, g_4, g_5} be the reference set of students.
- Let Ã be the fuzzy set of "smart" students, where "smart" is fuzzy term.

$$\tilde{A} = \{(g_1,0.4)(g_2,0.5)(g_3,1)(g_4,0.9)(g_5,0.8)\}$$

Here Ã indicates that the smartness of g_1 is 0.4 and so on

Fuzzy Sets (Continue)

Membership Function

- The membership function fully defines the fuzzy set
- A membership function provides a measure of *the degree of similarity* of an element to a fuzzy set

Membership functions can

- either be chosen by the user arbitrarily, based on the user's experience (MF chosen by two users could be different depending upon their experiences, perspectives, etc.)
- Or be designed using machine learning methods (e.g., artificial neural networks, genetic algorithms, etc.)

Fuzzy Sets (Continue)

There are different shapes of membership functions;

- **Triangular,**
- **Trapezoidal,**
- **Gaussian, etc**

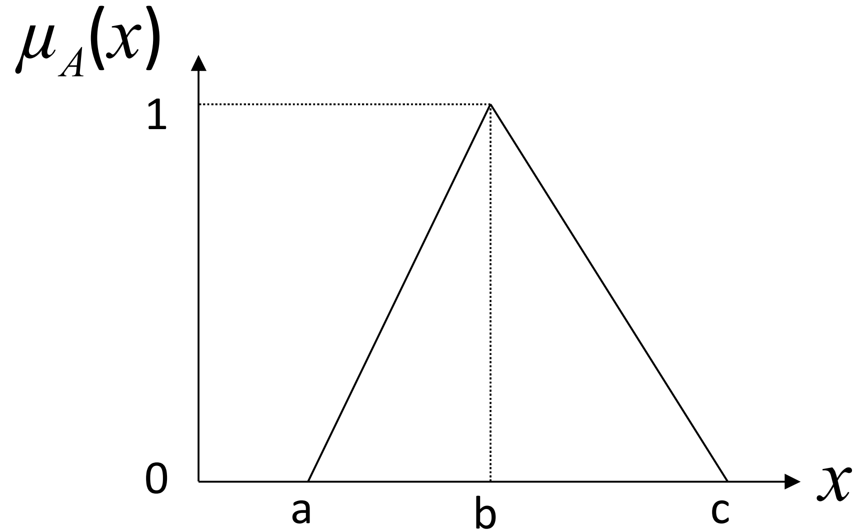
Fuzzy Sets (Continue)

- **Triangular membership function**

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

  a, b and c represent the *x* coordinates of the three vertices of $\mu_A(x)$ in a fuzzy set A (a: lower boundary and c: upper boundary where membership degree is zero, b: the centre where membership degree is 1)

$$\mu_A(x) = \begin{cases} 0 & \text{if } x \le a \\ \dfrac{x-a}{b-a} & \text{if } a \le x \le b \\ \dfrac{c-x}{c-b} & \text{if } b \le x \le c \\ 0 & \text{if } x \ge c \end{cases}$$



Fuzzy Sets (Continue)

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

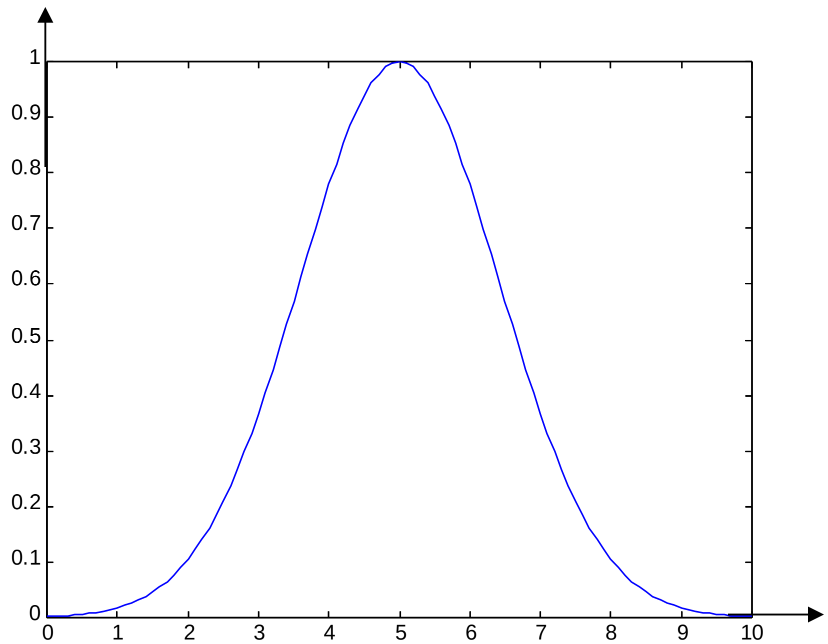
$$\mu_A(x) = \left\{ \begin{bmatrix} 0 & if\ x \le a \\ \dfrac{x-a}{b-a} & if\ a \le x \le b \\ 1 & if\ b \le x \le c \\ \dfrac{d-x}{d-c} & if\ c \le x \le d \\ 0 & if\ d \le x \end{bmatrix} \right\}$$

- **Gaussian membership function**

$$\mu_A(x,c,s,m) = \exp\left[-\frac{1}{2}\left|\frac{x-c}{s}\right|^m\right]$$

  - *c:* centre
  - *s:* width
  - *m:* fuzzification factor (e.g., *m*=2)

$\mu_A(x)$

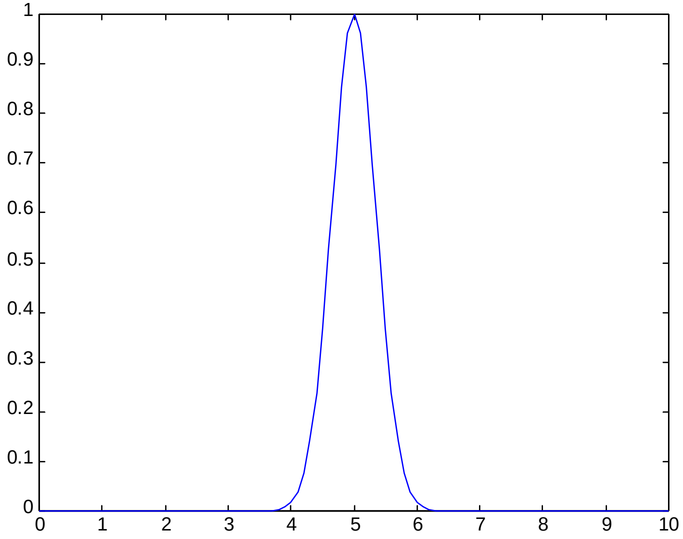


x

$c=5$

$s=2$

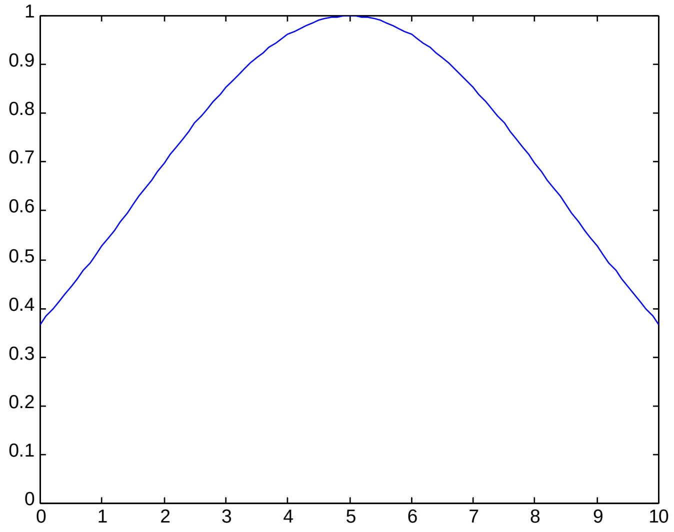
$m=2$



$c=5$

$s=0.5$

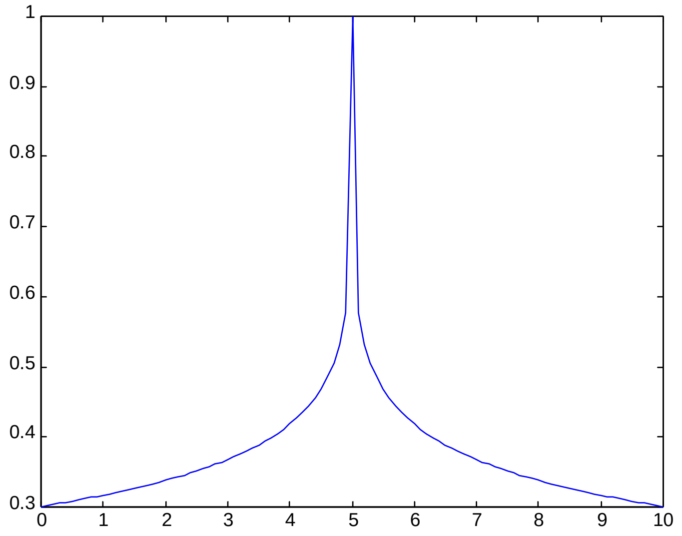
$m=2$



$c=5$

$s=5$

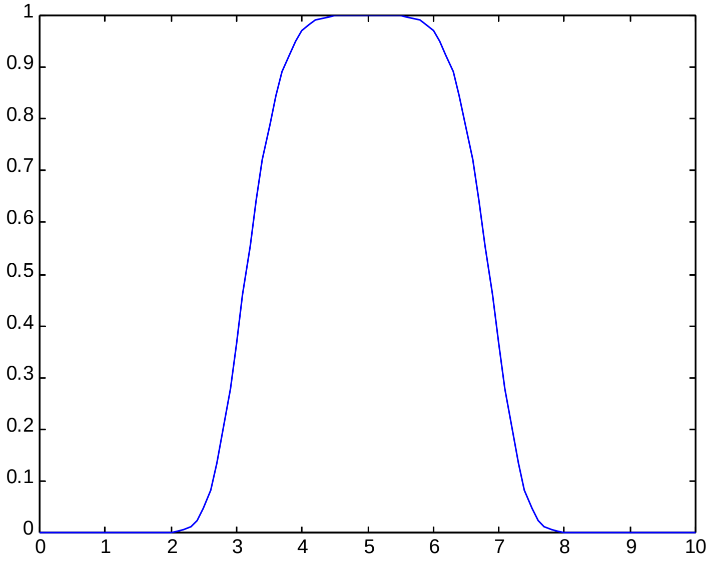
$m=2$



$c=5$

$s=2$

$m=0.2$



$c=5$

$s=5$

$m=5$

Fuzzy Set Operation

Given X to be the universe of discourse and Ã and Ḃ to be fuzzy sets with $\mu_A(x)$ and $\mu_B(x)$ are their respective membership function, the fuzzy set operations are as follows:

Union:

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

Intersection:

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

Complement:

$$\mu_A(x) = 1 - \mu_A(x)$$

Fuzzy Set Operation (Continue)

Example:

$A = \{(x_1,0.5),(x_2,0.7),(x_3,0)\}$ $B = \{(x_1,0.8),(x_2,0.2),(x_3,1)\}$

Union:

$$A \cup B = \{(x_1,0.8),(x_2,0.7),(x_3,1)\}$$

Because

$$\mu_{A \cup B}(x_1) = \max(\mu_A(x_1), \mu_B(x_1))$$

$$= \max(0.5,0.8)$$

$$= 0.8$$

$$\mu_{A \cup B}(x_2) = 0.7 \text{ and } \mu_{A \cup B}(x_3) = 1$$

Fuzzy Set Operation (Continue)

Example:

$A = \{(x_1,0.5),(x_2,0.7),(x_3,0)\}$ $B = \{(x_1,0.8),(x_2,0.2),(x_3,1)\}$

<u>Intersection:</u>

$$A \cap B = \{(x_1,0.5),(x_2,0.2),(x_3,0)\}$$

Because

$$\mu_{A \cap B}(x_1) = \min(\mu_A(x_1), \mu_B(x_1))$$

$$= \max(0.5,0.8)$$

$$= 0.5$$

$$\mu_{A \cap B}(x_2) = 0.2 \text{ and } \mu_{A \cap B}(x_3) = 0$$

Fuzzy Set Operation (Continue)

Example:

$A = \{(x_1,0.5),(x_2,0.7),(x_3,0)\}$

<u>Complement:</u>

$$A^c = \{(x_1,0.5),(x_2,0.3),(x_3,1)\}$$

Because

$$\mu_A(x_1) = 1 - \mu_A(x_1)$$

$$= 1 – 0.5$$

$$= 0.5$$

$$\mu_A(x_2) = 0.3 \text{ and } \mu_A(x_3) = 1$$

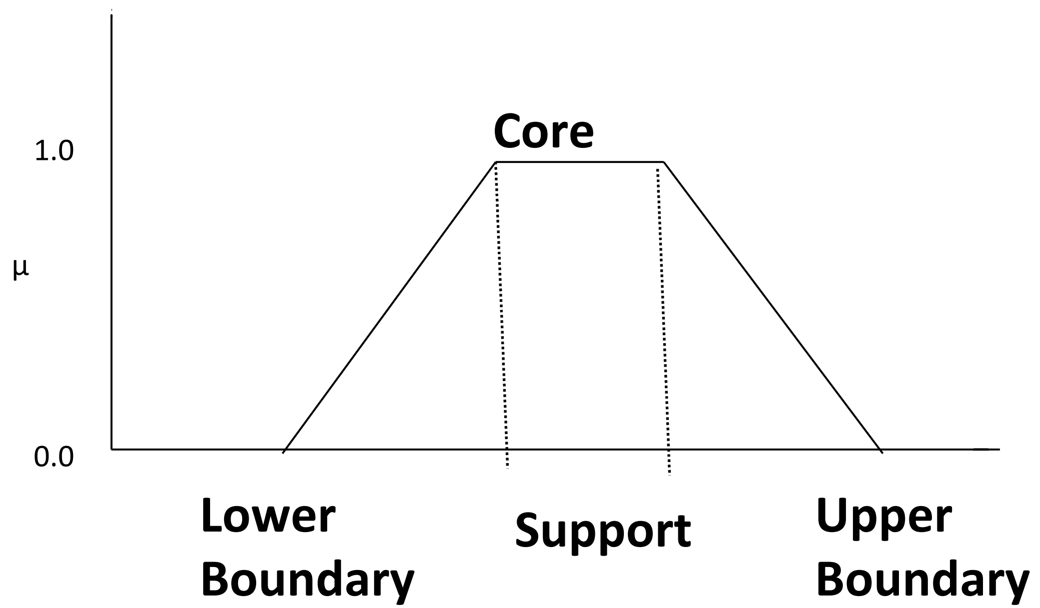
- **Support(A)** is set of all points x in X such that

$$\{(x|\ \mu_A(x) > 0\ \}$$

- **core(A)** is set of all points x in X such that

$$\{(x|\ \mu_A(x) =1\ \}$$

- Fuzzy set whose support is a single point in X with $\mu_A(x) =1$ is called fuzzy singleton


Core
1.0
μ
0.0
Lower
Boundary
Support
Upper
Boundary

Linguistic variable, linguistic term

- **Linguistic variable: A** *linguistic variable* is a variable whose values are 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.*
- *Tall is a linguistic* value or primary term

- 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,…}.*

Fuzzy 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* and “y is B” is called *consequent*.

Examples, for such a rule are

- If pressure is high, then volume is small.
- If the road is slippery, then driving is dangerous.
- If the fruit is ripe, then it is soft.

Binary fuzzy relation

- A binary fuzzy relation is a fuzzy set in $X \times Y$ which maps each element in $X \times Y$ to a membership value between *0* and *1*.
- If X and Y are two universes of discourse, then
- $R = \{((x,y), \mu_R(x, y)) \mid (x,y) \in X \times Y\}$ is a binary fuzzy relation in $X \times Y$.
- $X \times Y$ indicates cartesian product of X and Y

- The fuzzy rule *"If x is A then y is B"* may be abbreviated as $A \rightarrow B$ and is interpreted as $A \times B$.

- A fuzzy if then rule may be defined (Mamdani) as a binary fuzzy relation R on the product space $X \times Y$.

- $R = A \rightarrow B = A \times B = \int_{X \times Y} \mu_A(x)\ T\text{-}norm\ \mu_B(y) / (x,y)$.

expert systems: Fuzzy inference

Mamdani fuzzy inference

Sugeno fuzzy inference

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..

Fuzzy inference

Mamdani fuzzy inference

- The Mamdani-style fuzzy inference process is performed in four steps:
- Fuzzification of the input variables,
- Rule evaluation;
- Aggregation of the rule outputs, and finally
- Defuzzification.

Fuzzy inference

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

Rule 1:
IF x is A3
OR y is B1
THEN z is C1

Rule 2:
IF x is A2
OR y is B2
THEN z is C2

Rule 3:
IF x is A1
THEN z is C3

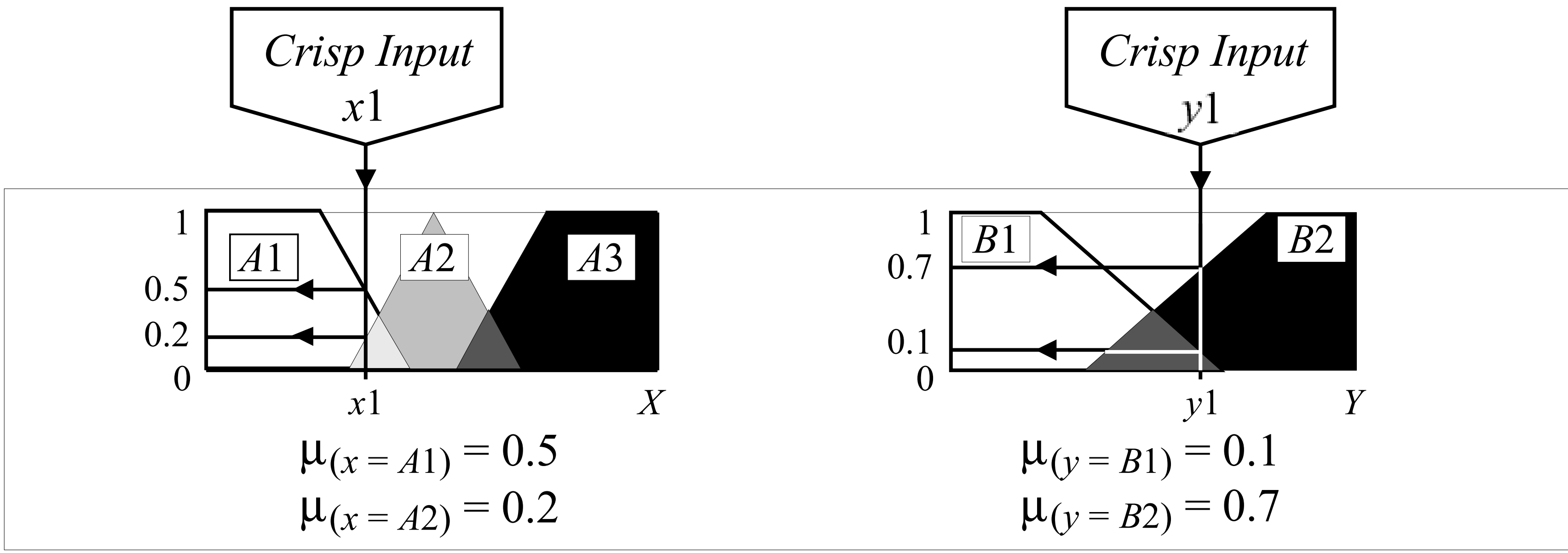
Rule 1:
IF *project_funding is enough*
OR *project_staffing is small*
THEN *risk is low*

Rule 2:
IF *project_funding is medium*
OR *project_staffing is large*
THEN *risk is normal*

Rule 3:
IF *project_funding is notenough*
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 $\mu_{(y=B\ 2)} = 0.7$,

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 (the truth value) is then applied to the consequent membership function.

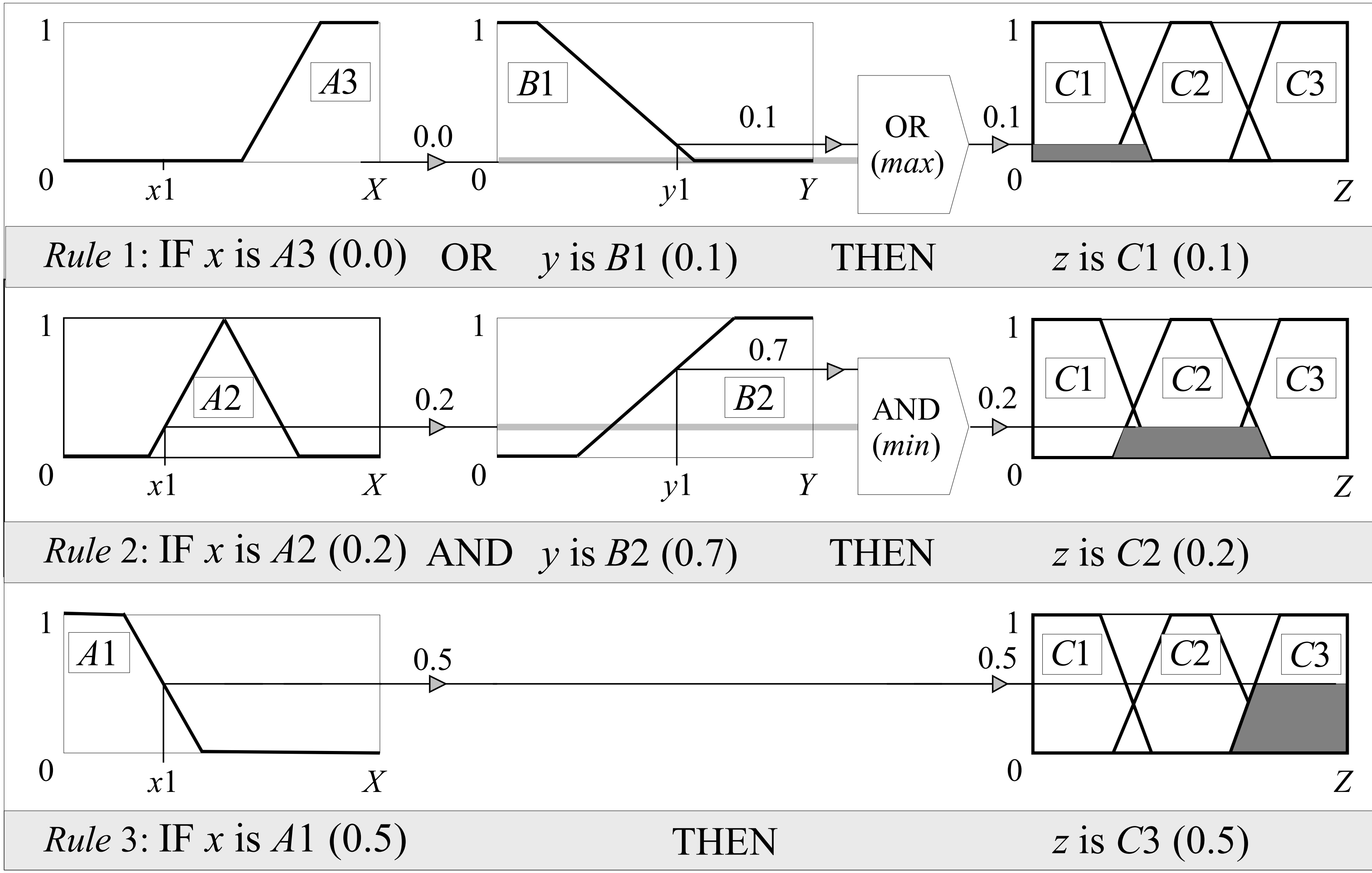
To evaluate the disjunction of the rule antecedents, we use the **OR fuzzy operation**. Typically, fuzzy expert systems make use of the classical fuzzy operation **union**:

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

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

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

Mamdani-style rule evaluation



Sugeno fuzzy inference

- **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:

$\boldsymbol{\mu_A(x) = 1}$ is called **fuzzy singleton**

- 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.

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

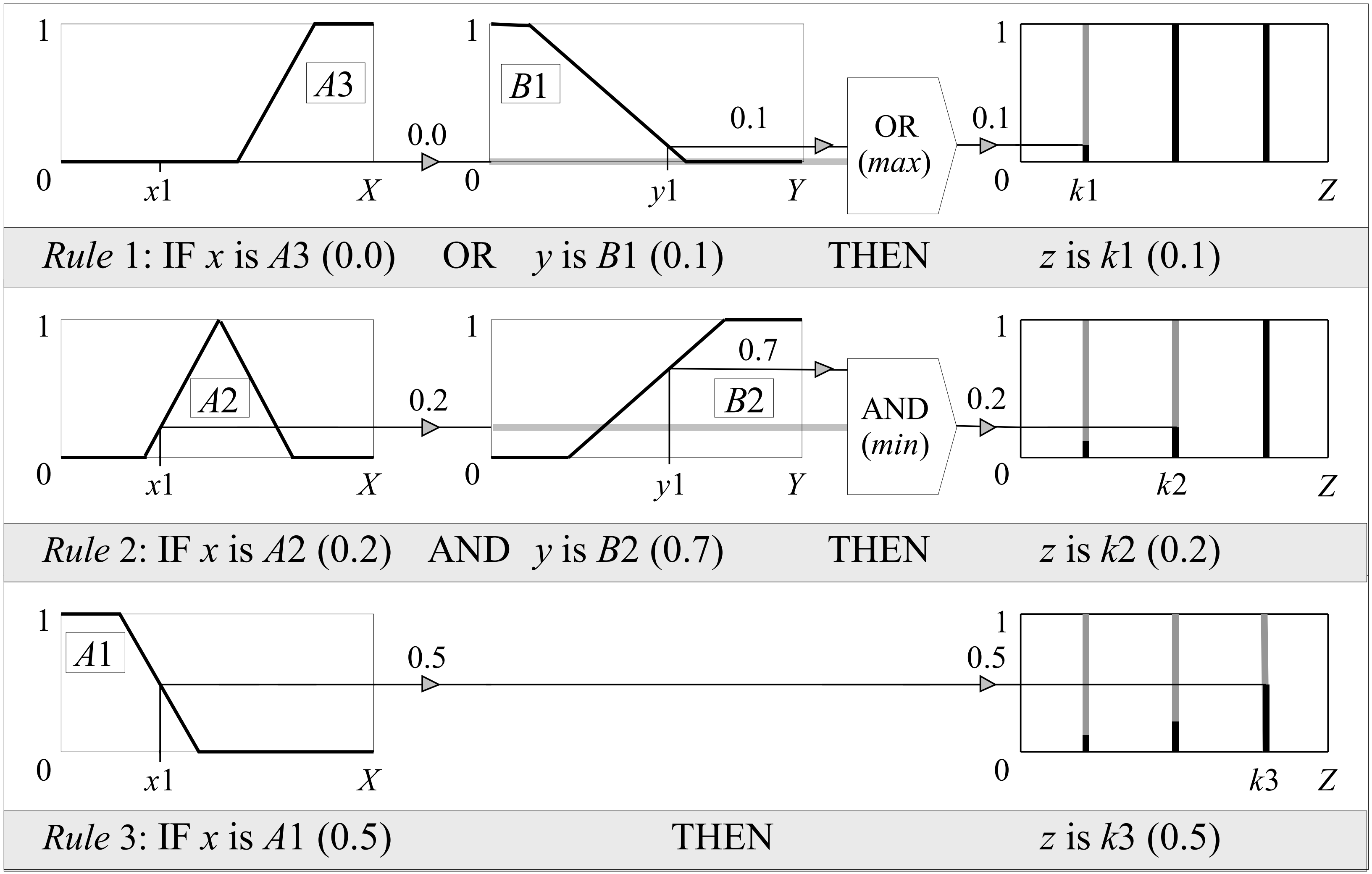
IF x is A
AND y is B
THEN z is k

where k is a constant.

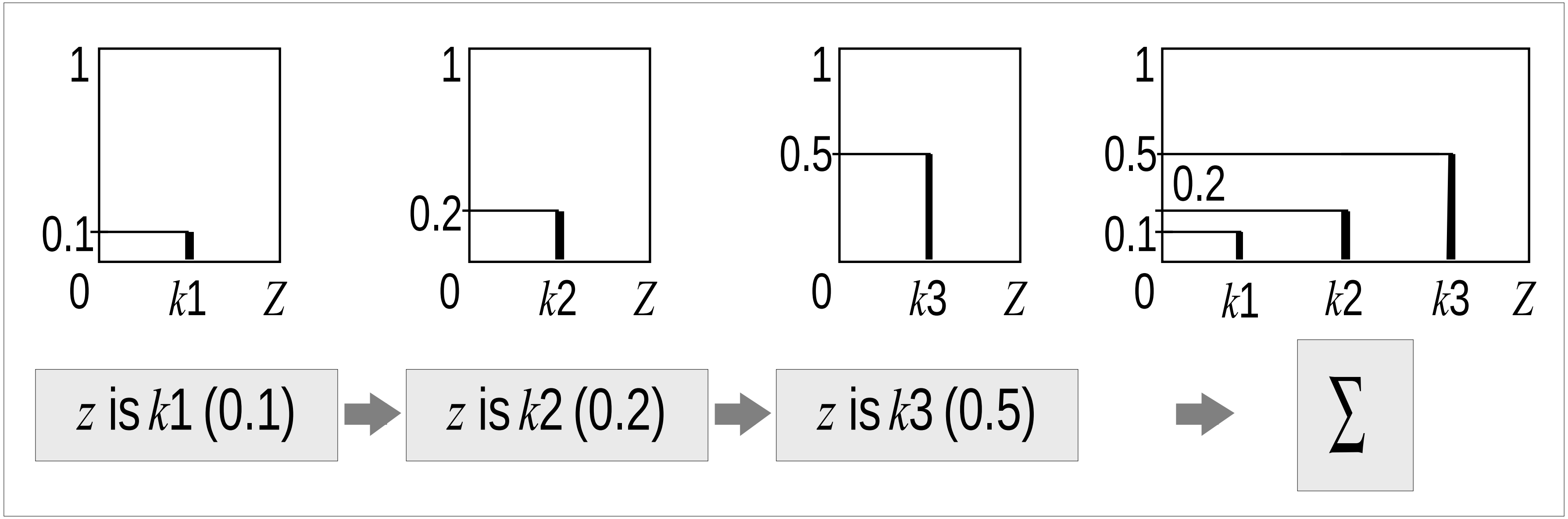
In this case, the output of each fuzzy rule is constant.

All resultant membership functions are represented by singleton spikes.

Sugeno-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