# Artificial Intelligence & Applications
# Code: 14B17CI772

Banerjee P. S.
Dept of CSE
JUET Guna

## Introduction

# Persons involved

- Faculty instructor 1: Dr. Banerjee P. S.  [BX1, BX2, BY2]
- Faculty instructor 2: Dr. Amit Rathi [BY1]

# Resources

- Main Text:
  - AI: Rich & Knight
- Other Main References:
  - AI & Expert Systems: Patterson
  - AI: Saroj Kaushik
  - Artificial Intelligence: A Modern Approach by Russell & Norvik, Pearson.
  - Principles of AI: Nilsson
  - Knowledge Based Systems: Mark Stefik
- Lab Main Text:
  - Logic and Prolog Programming: Saroj Kaushik
- Journals
  - AI, AI Magazine, IEEE Expert,
  - Area Specific Journals e.g, Computational Linguistics
- Conferences
  - IJCAI, AAAI

BOOK READING IS STRONGLY RECOMMENDED

SLIDES ARE JUST AN AID

## Intelligence

- Relate to tasks involving higher mental processes.

  Examples:

  creativity, solving problems, pattern recognition, classification, learning, induction, deduction, building analogies, optimization, language processing, knowledge and many more.

- Intelligence is the computational part of the ability to achieve goals.

## Intelligent Behavior

- **Perceiving** one's environment,
- **Acting** in complex environments,
- **Learning** and understanding from experience,
- **Reasoning** to solve problems and discover hidden knowledge,
- **Knowledge** applying successfully in new situations,
- **Thinking** abstractly, using analogies,
- **Communicating** with others, and more like
- **Creativity**, **Ingenuity**, **Expressive-ness, Curiosity.**

## What is Artificial Intelligence ?

- John McCarthy, who coined the term Artificial Intelligence in 1956, defines it as "the science and engineering of making intelligent machines", especially intelligent computer programs.
- Artificial Intelligence (AI) is the intelligence of machines and the branch of computer science that aims to create it.
- Intelligence is the computational part of the ability to achieve goals in the world. Varying kinds and degrees of intelligence occur in people, many animals and some machines.
- AI is the study of the mental faculties through the use of computational models.
- AI is the study of : How to make computers do things which, at the moment, people do better.
- AI is the study and design of intelligent agents, where an intelligent agent is a system that perceives its environment and takes actions that maximize its chances of success.

## Artificial Intelligence (AI)

### The definitions of AI outlined in textbooks

**(a)** 'The exciting new effort to make computers think ... machines with minds, in the full and literal sense' (Haugeland, 1985)

'The automation of activities that we associate with human thinking, activities such as decision-making, problem solving, learning ...' (Bellman, 1978)

**(b)** 'The study of mental faculties through the use of computational models' (Charniak and McDermott, 1985)

'The study of the computations that make it possible to perceive, reason, and act' (Winston, 1992)

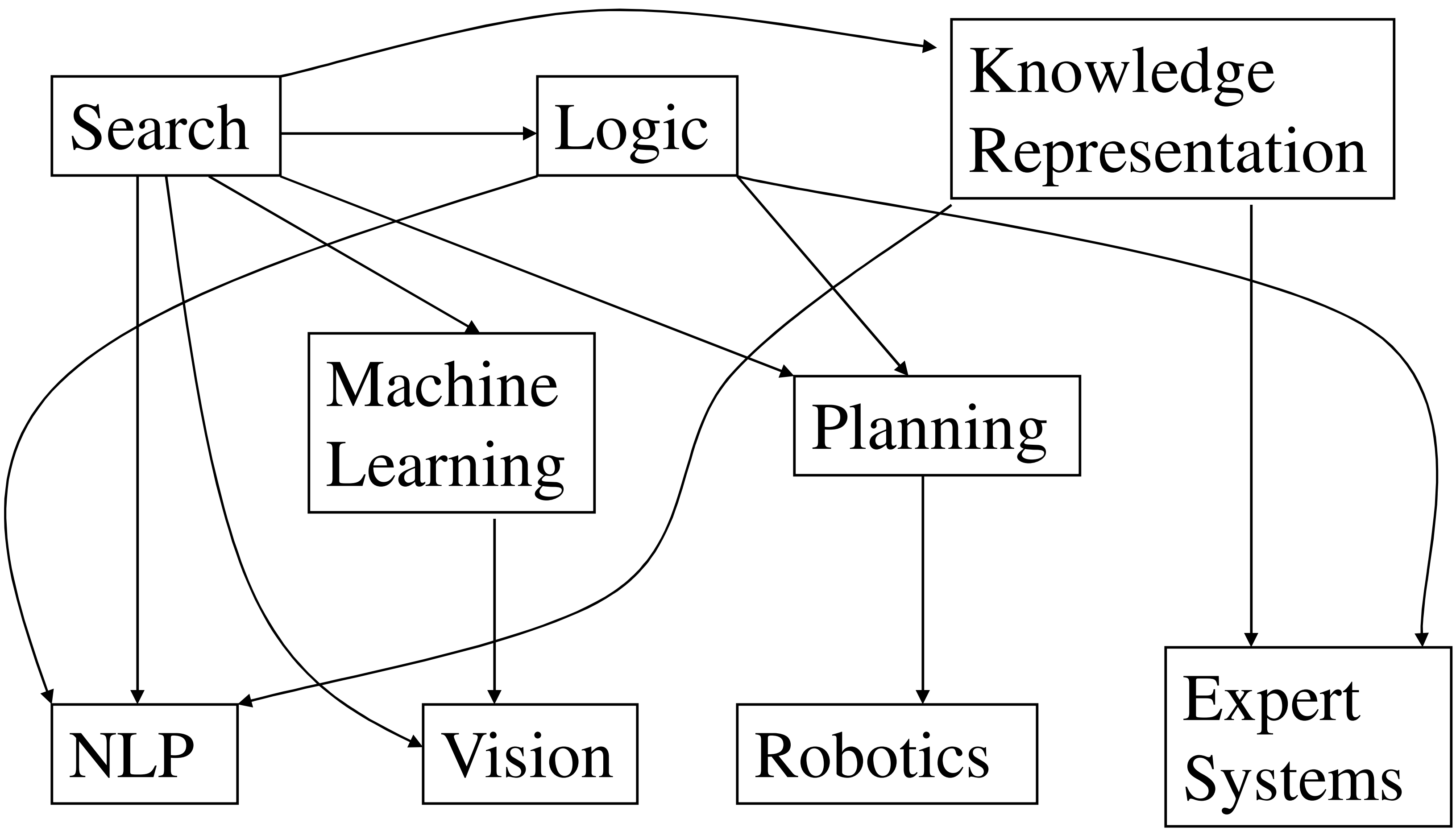
**(c)** 'The art of creating machines that perform functions that require intelligence when performed by people' (Kurzweil, 1990)

'The study of how to make computers do things at which, at the moment, people are better' (Rich and Knight, 1991)

**(d)** 'A field of study that seeks to explain and emulate intelligent behavior in terms of computational processes' (Schalkoff, 1990)

'The branch of computer science that is concerned with the automation of intelligent behavior' (Luger and Stubblefield, 1993)

# Areas of AI and their inter-dependencies



# Allied Disciplines

| | |
|---|---|
| Philosophy | Knowledge Rep., Logic, Foundation of AI (is AI possible?) |
| Maths | Search, Analysis of search algos, logic |
| Economics | Expert Systems, Decision Theory, Principles of Rational Behavior |
| Psychology | Behavior Based insights into AI programs |
| Brain Science | Learning, Neural Nets |
| Physics | Learning, Information Theory & AI, Entropy, Robotics |
| Computer Sc. & Engg. | Systems for AI |

## Understanding AI

- How knowledge is acquired, represented, and stored;
- How intelligent behavior is generated and learned;
- How motives, emotions, and priorities are developed and used;
- How sensory signals are transformed into symbols;
- How symbols are manipulated to perform logic, to reason about past, and plan for future;
- How mechanisms of intelligence produce the phenomena of illusion, belief, hope, fear, dreams, kindness and love."

## Hard or Strong AI

- Generally, artificial intelligence research aims to create AI that can **replicate human intelligence** completely.

- Strong AI refers to a machine that approaches or supersedes human intelligence,
  - If it can do typically human tasks,
  - If it can apply a wide range of background knowledge and
  - If it has some degree of self-consciousness.

- Strong AI aims to build machines whose overall intellectual ability is indistinguishable from that of a human being.

## Soft or Weak AI

- Weak AI refers to the use of software to study or accomplish specific problem solving or reasoning tasks that do not encompass the full range of human cognitive abilities.

- Example : a chess program such as **Deep Blue.**

- Weak AI does not achieve self-awareness; it demonstrates wide range of human-level cognitive abilities; it is merely an intelligent, a specific problem-solver.

## General AI Goal

- Replicate human intelligence : still a distant goal.
- Solve knowledge intensive tasks.
- Make an intelligent connection between perception and action.
- Enhance human-human, human-computer and computer to computer interaction / communication.

**Engineering based AI Goal**

- Develop concepts, theory and practice of building intelligent machines
- Emphasis is on system building.

**Science based AI Goal**

- Develop concepts, mechanisms and vocabulary to understand biological intelligent behavior.
- Emphasis is on understanding intelligent behavior.

# Goals of AI

- The definitions of AI gives four possible goals to pursue :

  1. Systems that think like humans.
  2. Systems that think rationally.
  3. Systems that act like humans
  4. Systems that act rationally

- Traditionally, all four goals have been followed and the approaches were:

| | Human-like | Rationally |
|---|---|---|
| **Think** | **(1) Cognitive science Approach** | **(2) Laws of thought Approach** |
| **Act** | **(3) Turing Test Approach** | **(4) Rational agent Approach** |

- Most of AI work falls into category (2) and (4).

# AI Approaches

The approaches followed are defined by choosing goals of the computational model, and basis for evaluating performance of the system.

## 3.1 Cognitive science : Think human-like

- An exciting new effort to make computers think; that it is, the machines with minds, in the full and literal sense.
- Focus is not just on behavior and I/O, but looks at reasoning process.
- Computational model as to how results were obtained.
- Goal is not just to produce human-like behavior but to produce a sequence of steps of the reasoning process, similar to the steps followed by a human in solving the same task.

## Cognitive Science

- Aims to develop, explore and evaluate theories of how the mind works through the use of computational models.
- The important is not what is done but how it is done; means intelligent behavior is not enough, the program must operate in an intelligent manner.
- Example : The Chess programs are successful, but say little about the ways humans play chess.

- Cognitive science is the scientific study of the human mind. It is a highly interdisciplinary field, combining ideas and methods from psychology, computer science, linguistics, philosophy, and neuroscience.

- The broad goal of cognitive science is to characterize the nature of human knowledge – its forms and content – and how that knowledge is used, processed, and acquired.

- Active areas of cognitive research in the Department include language, memory, visual perception and cognition, thinking and reasoning, social cognition, decision making, and cognitive development. [BCS = Brain + Cognitive Science]

#https://bcs.mit.edu/research/cognitive-science

Rules:

Much of human knowledge is naturally described in terms of rules of the form IF … THEN …, and many kinds of thinking such as planning can be modeled by rule-based systems. The explanation schema used is:

*Explanation target:*
- Why do people have a particular kind of intelligent behavior?

*Explanatory pattern:*

i. People have mental rules.
ii. People have procedures for using these rules to search a space of possible solutions, and procedures for generating new rules.
iii. Procedures for using and forming rules produce the behavior.

Concepts:

Concepts, which partly correspond to the words in spoken and written language, are an important kind of mental representation. There are computational and psychological reasons for abandoning the classical view that concepts have strict definitions.

*Explanatory target:* Why do people have a particular kind of intelligent behavior?

*Explanation pattern:*

I. People have a set of concepts, organized via slots that establish kind and part hierarchies and other associations.
II. People have a set of procedures for concept application, including spreading activation, matching, and inheritance.
III. The procedures applied to the concepts produce the behavior.
IV. Concepts can be translated into rules, but they bundle information differently than sets of rules, making possible different computational procedures.

Analogies:

Analogies play an important role in human thinking, in areas as diverse as problem solving, decision making, explanation, and linguistic communication. Computational models simulate how people retrieve and map source analogs in order to apply them to target situations. The explanation schema for analogies is:

*Explanation target:*

•Why do people have a particular kind of intelligent behavior?

*Explanatory pattern:*

•People have verbal and visual representations of situations that can be used as cases or analogs.

•People have processes of retrieval, mapping, and adaptation that operate on those analogs.

•The analogical processes, applied to the representations of analogs, produce the behavior.

#https://plato.stanford.edu/entries/cognitive-science/#Met

## Laws of Thought : Think Rationally

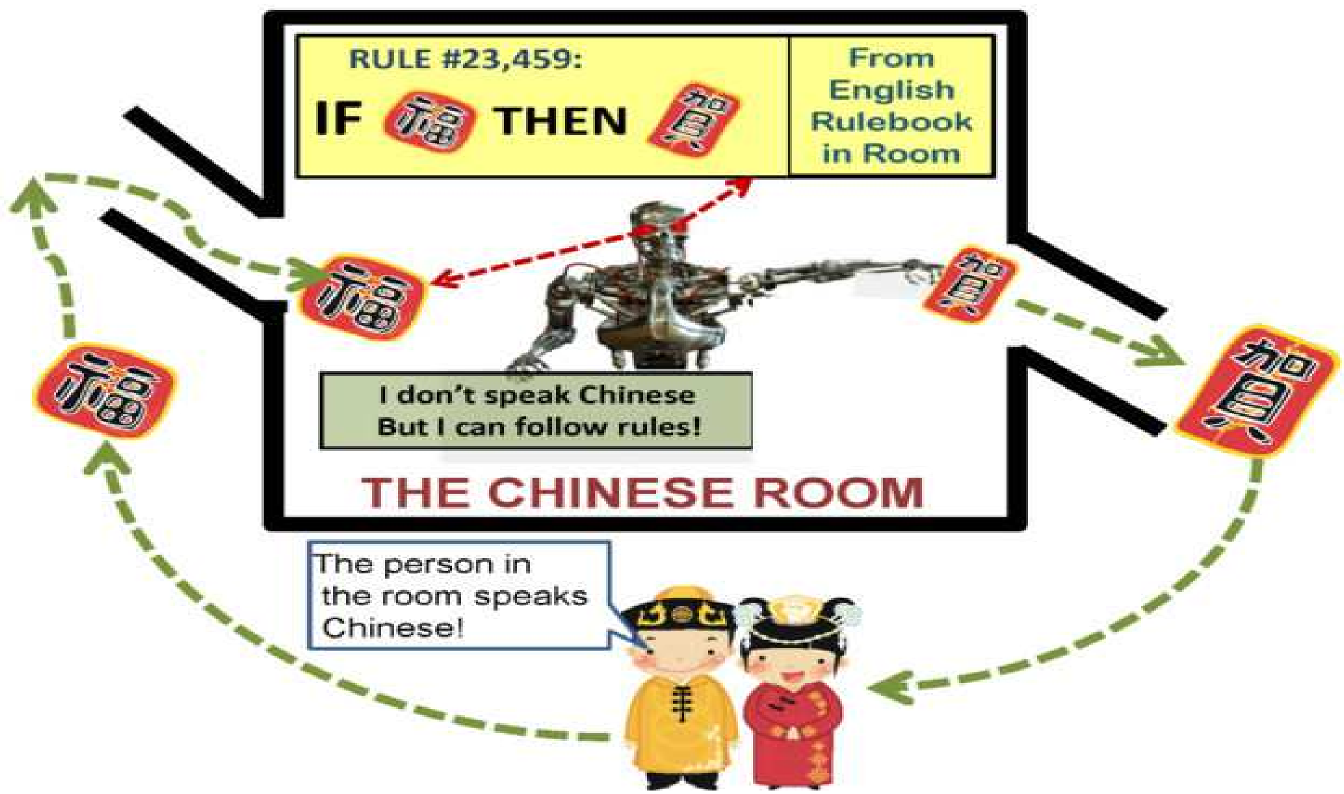
- The study of mental faculties through the use of computational models; that it is, the study of the computations that make it possible to perceive, reason, and act.
- Focus is on inference mechanisms that are provably correct and guarantee an optimal solution.
- Develop systems of representation to allow inferences to be like *"Socrates is a man. All men are mortal. Therefore Socrates is mortal."*
- Goal is to formalize the reasoning process as a system of logical rules and procedures for inference.
- The issue is, not all problems can be solved just by reasoning and inferences.

## Turing Test : Act Human-like

- The art of creating machines that perform functions requiring intelligence when performed by people; that it is the study of, how to make computers do things which at the moment people do better.
- Focus is on action, and not intelligent behavior centered around representation of the world.
- A Behaviorist approach, is not concerned with how to get results but to the similarity to what human results are.
- **Example : Turing Test**
  - 3 rooms contain: a person, a computer, and an interrogator.
  - The interrogator can communicate with the other 2 by teletype (to avoid the machine imitate the appearance or voice of the person).
  - The interrogator tries to determine which is the person and which is the machine.
  - The machine tries to fool the interrogator to believe that it is the human, and the person also tries to convince the interrogator that it is the human.
  - If the machine succeeds in fooling the interrogator, then conclude that the machine is intelligent.
- Goal is to develop systems that are human-like.

# Challenge to Turing Test

- Chinese Room Argument
- Problem of automated translation



# ✓Translation and Cognition

## Rational Agent : Act Rationally

- Tries to explain and emulate intelligent behavior in terms of computational processes; that it is concerned with the automation of intelligence.
- Focus is on systems that act sufficiently if not optimally in all situations;
- It is passable to have imperfect reasoning if the job gets done.
- Goal is to develop systems that are rational and sufficient.

# AI Techniques

Various techniques that have evolved, can be applied to a variety of AI tasks. The techniques are concerned with how we represent, manipulate and reason with knowledge in order to solve problems.

**Example**

- Techniques, not all "intelligent" but used to behave as intelligent
  - Describe and match
  - Constraint satisfaction
  - Generate and test
  - Goal reduction
  - Tree Searching
  - Rule based systems
- Biology-inspired AI techniques are currently popular
  - Neural Networks
  - Reinforcement learning
  - Genetic Algorithms

## DESCRIBE AND MATCH

I. Suited to problems where we wish to identify rules which describe transitions from one state to another.

II. Representation must include start and end state descriptions and a set of possible "transition rules" that might be applied.

III. Problem is then to find the appropriate transition rules.

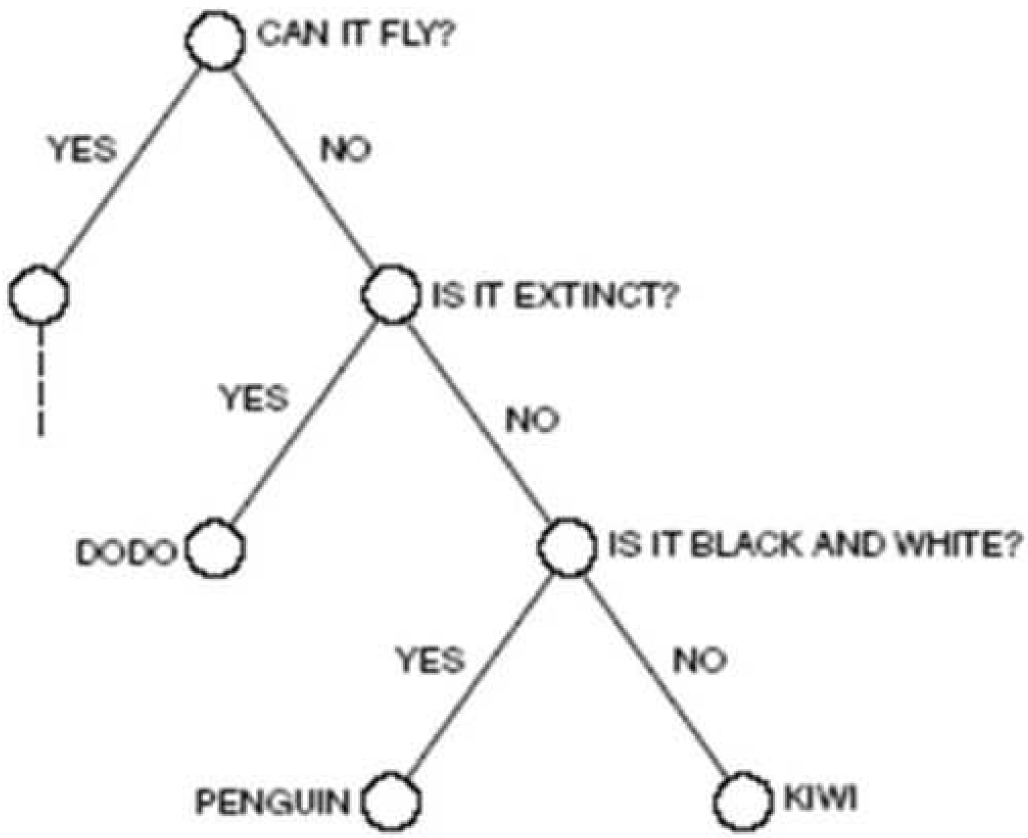
IV. Describe and match problems are generally addressed using *goal-reduction* techniques.

## Techniques that make system to behave as "Intelligent"

### Describe and Match

- **Model** is a description of a system's behavior.
- **Finite state model** consists of a set of states, a set of input events and the relations between them. Given a current state and an input event you can determine the next current state of the model.
- **Computation model** is a finite state machine. It includes of a set of states, *a set of* start states, *an input* alphabet, *and a* transition function *which* maps input symbols and current states to a *next state*.
- **Representation of computational system** include start and end state descriptions and a set of possible transition rules that might be applied. Problem is to find the appropriate transition rules.
- **Transition relation:** If a pair of states **(S, S')** is such that one move takes the system from **S** to **S'**, then the transition relation is represented by **S => S'**
- **State-transition system** is called deterministic if every state has at most one successor; it is called non-deterministic if at least one state has more than one successor.
- Examples of some possible transitions between states are shown for the Towers of Hanoi puzzle.

# Describe and Match Search Tree Representation to identify Penguin

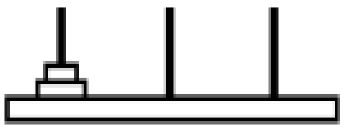


# Describe and Match Search Tree Representation to Identify a Tiger and a Cat

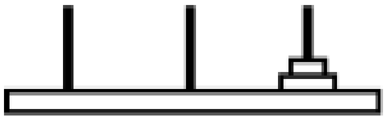
- **Puzzle : Towers of Hanoi with only 2 disks**

**Solve the puzzle :**

Initial state



Goal state

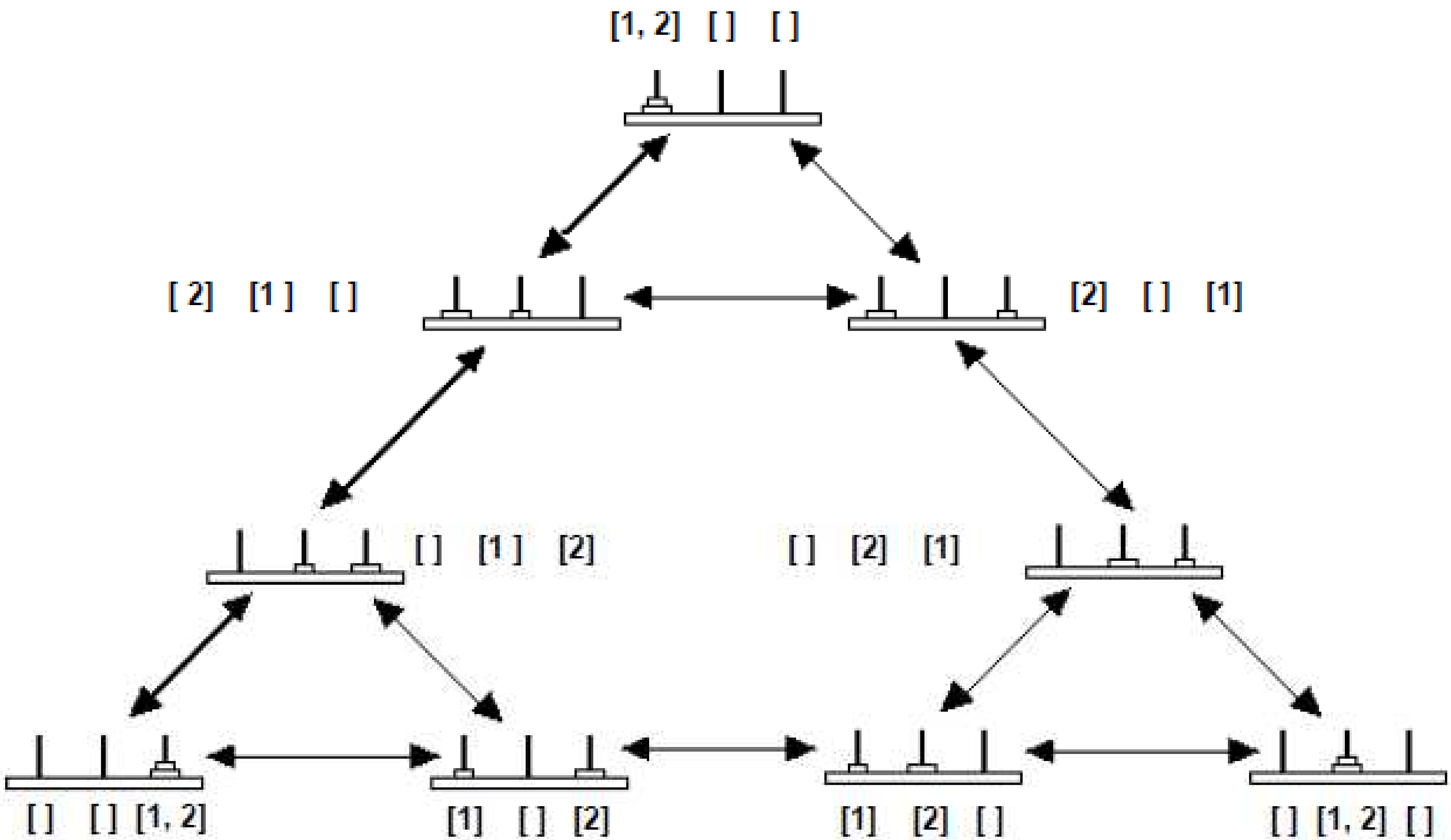


Move the disks from the leftmost post to the rightmost post while

- never putting a larger disk on top of a smaller one;
- move one disk at a time, from one peg to another;
- middle post can be used for intermediate storage.

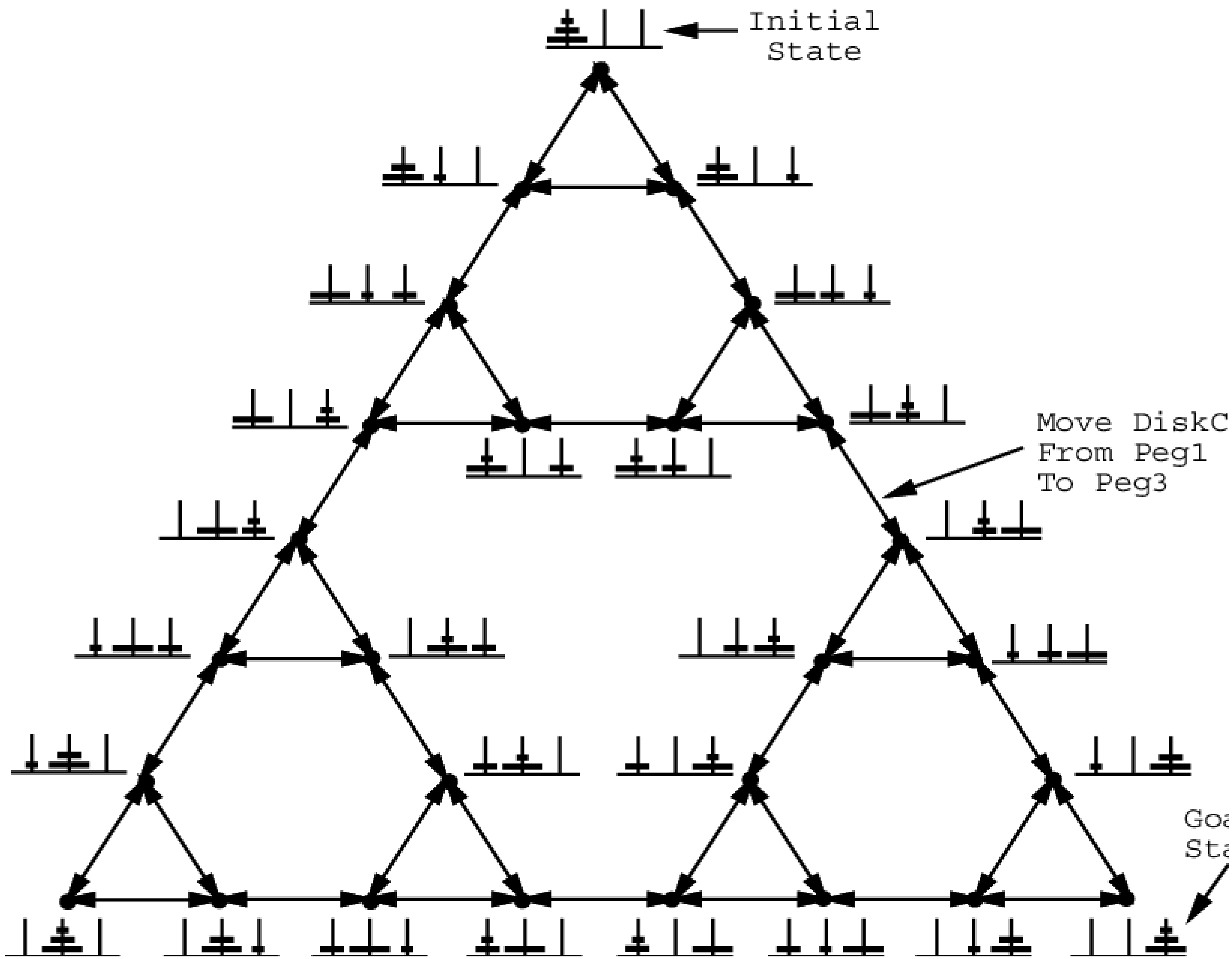
**Play the game in the smallest number of moves possible.**

- Possible state transitions in the Towers of Hanoi puzzle with 2 disks.




Towers of Hanoi puzzle

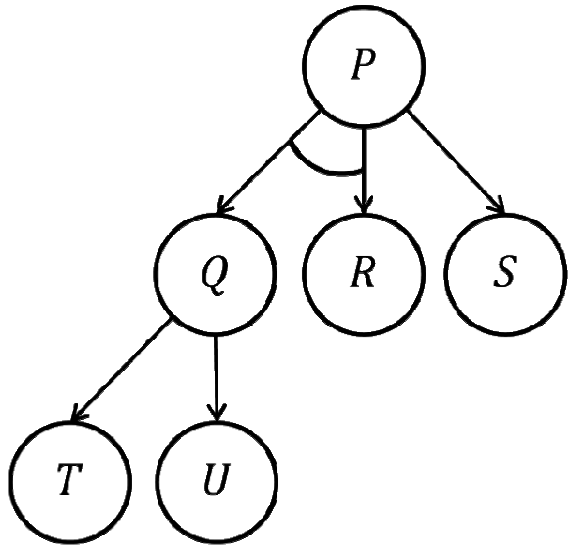
- **Shortest solution** is the sequence of transitions from the top state downward to the lower left.


Initial
State
Move DiskC
From Peg1
To Peg3
Goa
Sta

## Goal Reduction

- Goal-reduction procedures are a special case of the procedural representations of knowledge in AI; an alternative to declarative, logic-based representations.
- The process involves the hierarchical sub-division of goals into sub-goals, until the sub-goals which have an immediate solution are reached and said "goal has been satisfied".
- Goal-reduction process is illustrated in the form of AND/OR tree drawn upside-down.
  - **Goal levels** : Higher-level goals are higher in the tree, and lower-level goals are lower in the tree.
  - **Arcs** are directed from a higher-to-lower level node represents the reduction of higher-level goal to lower-level sub-goal.
  - **Nodes** at the bottom of the tree represent irreducible action goals.
- An AND-OR tree/graph structure can represent relations between goals and sub-goals, alternative sub-goals and conjoint sub-goals.

# AND-OR Tree/Graph Techniques

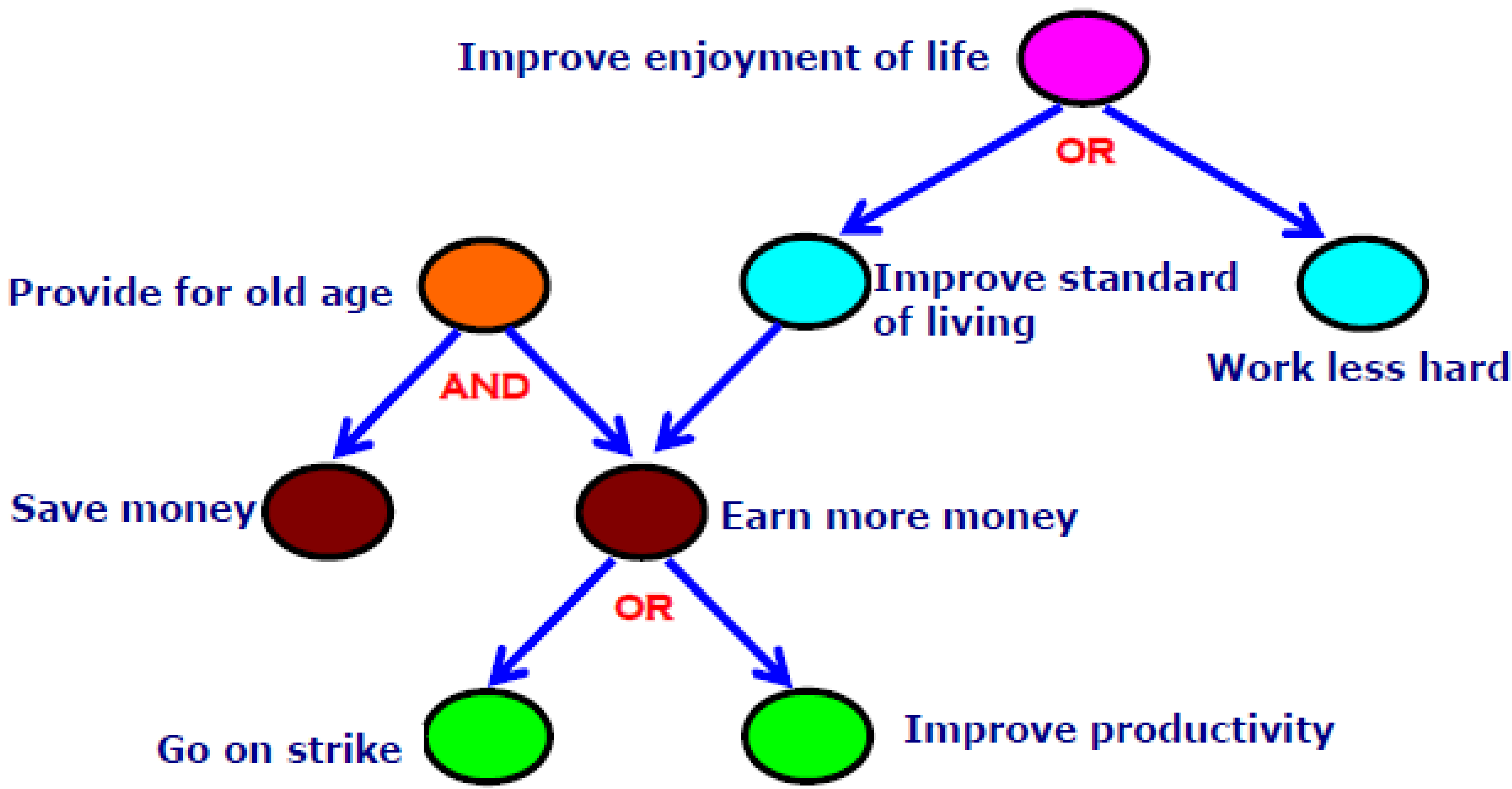



The and-or tree represents the search space for solving the problem P, using the goal-reduction methods:

✓P if Q and R
✓P if S
✓Q if T
✓Q if U

- **Example** Goal Reduction

AND-OR tree/graph structure to represent facts such as "enjoyment", "earning/save money", "old age" etc.




AND-OR tree/graph structure

The above AND-OR tree/graph structure describes

- **Hierarchical relationships between goals and subgoals**

  The "going on strike" is a sub-goal of "earning more money", is a sub-goal of "improving standard of living ", is a sub-goal of "improving enjoyment of life".

- **Alternative ways of trying to solve a goal**

  The "going on strike" and "increasing productivity" are alternative ways of trying to "earn more money" (increase pay).

  e.g.: "improving standard of living" and "working less hard" are alternative ways of trying to "improve enjoyment of life".

- **Conjoint sub-goals**

  To "provide for old age", not only need to "earn more money", but as well need to "save money".

The and-or tree representation for the following Goals:

- ✓Enjoyment
- ✓Get placed
- ✓Study hard
- ✓Better internship
- ✓Have Reference
- ✓Good Coding Skills
- ✓Waste time

## Constraint Satisfaction Techniques

- Constraint is a logical relation among variables. e.g. *"circle is inside the square"* – The constraints relate objects without precisely specifying their positions; moving any one, the relation is still maintained.

- Constraint satisfaction is a process of finding a solution to a set of constraints – the constraints express allowed values for variables and finding solution is evaluation of these variables that satisfies all constraints.

## Constraint Satisfaction Problem (CSP) and its solution

- A Constraint Satisfaction Problem (CSP) consists of :
  - Variables, a finite set $X = \{x_1, \ldots, x_n\}$,
  - Domain, a finite set **Di** of possible values which each variable xi can take,
  - Constraints, a set of values that the variables can simultaneously satisfy the constraint (e.g. $D_1 \mathrel{!=} D_2$)

- A solution to a CSP is an *assignment of a value* from its domain to every variable satisfying every constraint; that could be :
  - one solution, with no preference as to which one,
  - all solutions,
  - an optimal, or a good solution - Constraint Optimization Problem (COP).

- Constraint satisfaction has application in Artificial Intelligence, Programming Languages, Symbolic Computing, Computational Logic.

- **Example 1 : N-Queens puzzle**

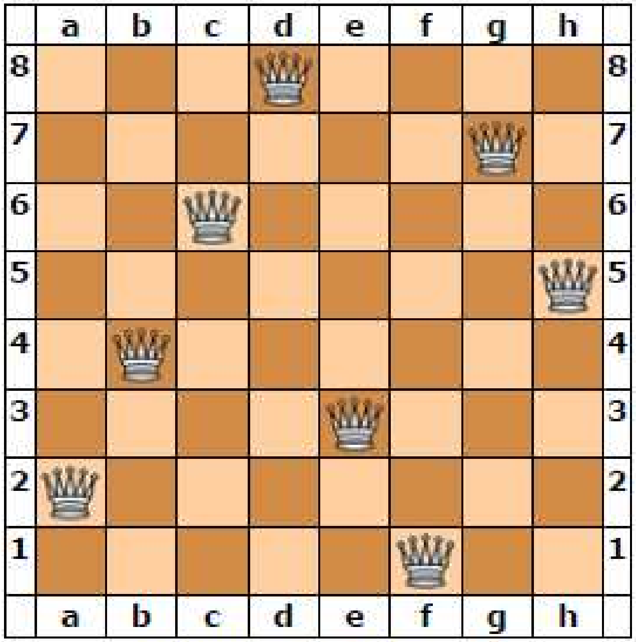
  Problem : Given any integer **N**, place **N** queens on **N*N** chessboard satisfying constraint that no two queens threaten each other.

  (a queen threatens other queens on same row, column and diagonal).

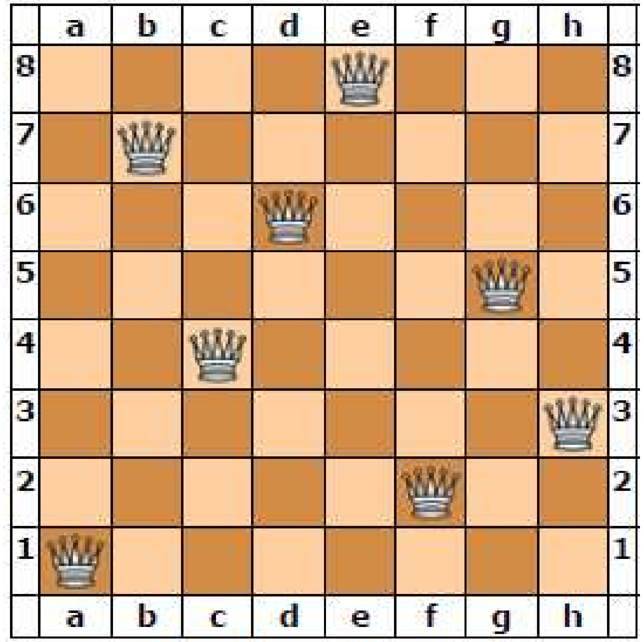
  Solution : To model this problem

  - Assume that each queen is in different column;
  - Assign a variable **Ri (i = 1 to N)** to the queen in the i-th column indicating the position of the queen in the row.
  - Apply "**no-threatening**" constraints between each couple $\mathbf{R_i}$ and $\mathbf{R_j}$ of queens and evolve the algorithm.

◇ Example : 8 - Queens puzzle




Unique solution 1



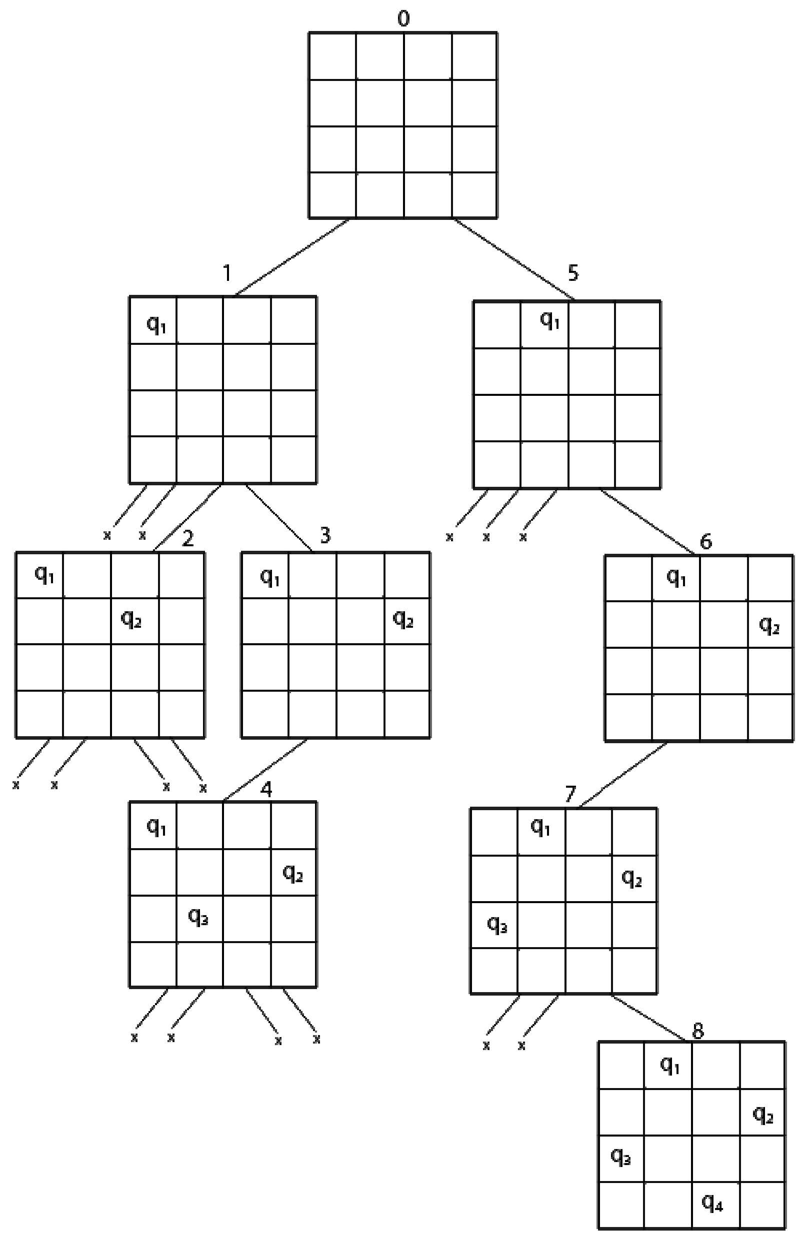

Unique solution 2

◇ The eight queens puzzle has 92 distinct solutions. If solutions that differ only by symmetry operations (rotations and reflections) of the board are counted as one. The puzzle has 12 unique solutions. Only two solutions are presented above.

try to place first queen
success
  try to place second queen
  success
    try to place third queen
    fail
  try to place second queen in another position
  success
    try to place third queen
    success
      try to place fourth queen

1) Start in the leftmost column
2) If all queens are placed
   return true
3) Try all rows in the current column.
   Do following for every tried row.
   a) If the queen can be placed safely in this row then mark this [row, column] as part of the solution and recursively check if placing queen here leads to a solution.
   b) If placing the queen in [row, column] leads to a solution then return true.
   c) If placing queen doesn't lead to a solution then unmark this [row, column] (Backtrack) and go to step (a) to try other rows.
3) If all rows have been tried and nothing worked, return false to trigger backtracking.

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | | | $q_1$ | |
| 2 | $q_2$ | | | |
| 3 | | | | $q_3$ |
| 4 | | $q_4$ | | |

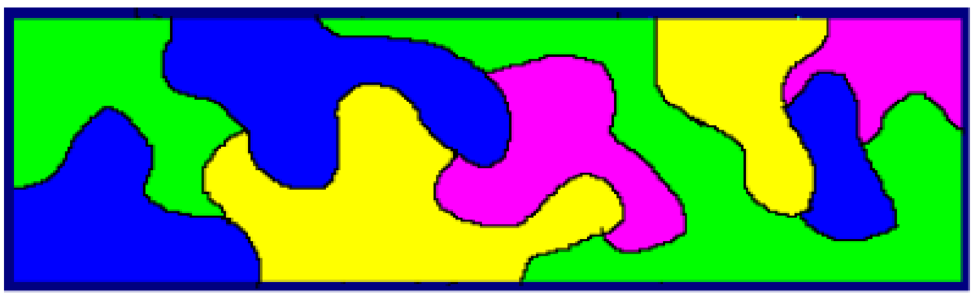

0
1
5
2
3
6
4
7
8
q1
q2
q3
q4
x

**Example 2 : Map Coloring**

Problem : Given a map (graph) and a number of colors, the problem is to assign colors to those areas in the map (nodes) satisfying the constraint that no adjacent nodes (areas) have the same color assigned to them.

Solution : To model this Map Coloring problem

- Label each node of the graph with a variable (domain corresponding to the set of colors);
- Introduce the non-equality constraint between two variables labeling adjacent nodes.
- A 4 – color map



- The "**Four Color Theorem**", states that 4 - colors are sufficient to color any map so that regions sharing a common border receive different colors.

## Tree Searching

- Many problems (e.g. goal reduction, constraint networks) can be described in the form of a search tree. A solution to the problem is obtained by finding a path through this tree.
- A search through the entire tree, until a satisfactory path is found, is called exhaustive search.
- **Tree search strategies:**

  - **Depth-first search**

    * Assumes any one path is as good as any other path.
    * At each node, pick an arbitrary path and work forward until a solution is found or a dead end is reached.
    * In the case of a dead end - backtrack to the last node in the tree where a previously unexplored path branches of, and test this path.
    * **Backtracking can be of two types :**
      - Chronological backtracking : undo everything as we move back "up" the tree to a suitable node.
      - Dependency directed backtracking : only withdraw choices that "matter" (ie those on which dead end depends).

**The four other types of search strategies are :**

◇ **Hill climbing**

* Like depth first but involving some quantitative decision on the "most likely" path to follow at each node.

◇ **Breadth-first search**

* Look for a solution amongst all nodes at a given level before proceeding to the next.

◇ **Beam search**

* Like breadth first (level by level) but selecting only those N nodes at each level that are "most likely" to lead to a solution.

◇ **Best-first search**

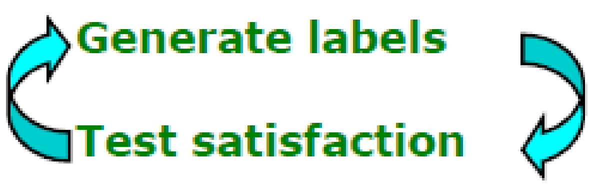
* Like beam search but only proceeding from one "most likely" node at each level.

## Generate and Test (GT)

- Most algorithms for solving Constrain Satisfaction Problems (CSPs) search systematically through the possible assignments of values.
  - ◇ CSP algorithms guarantee to find a solution, if one exists, or to prove that the problem is unsolvable.
  - ◇ the disadvantage is that they take a very long time to do so.

- **Generate-and-test method**

  The method first guesses the solution and then tests whether this solution is correct, means solution satisfies the constraints.
  - ◇ This paradigm involves two processes:
    - * Generator to enumerate possible solutions (hypotheses).
    - * Test to evaluate each proposed solution
  - ◇ The algorithm is



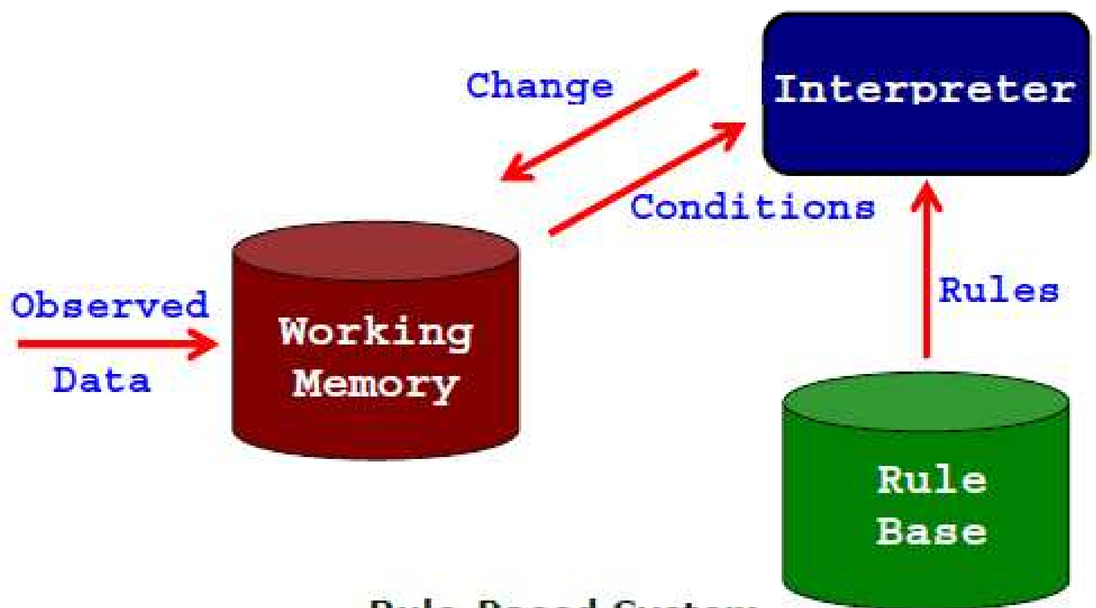
**Disadvantages**

- Not very efficient; generates many wrong assignments of values to variables which are rejected in the testing phase.
- Generator leaves conflicting instantiations and it generates other assignments independently of the conflict.
- For better efficiency GT approach need to be supported by backtracking approach.

**Example:** Opening a combination lock without knowing the combination.

## Rule-Based Systems (RBSs)

- Rule-based systems are simple and successful AI technique.
  - Rules are of the form: **IF <condition> THEN <action>**.
  - Rules are often arranged in hierarchies ("and/or" trees).
  - When all conditions of a rule are satisfied the rule is triggered.
- **RBS Components :** Working Memory, Rule Base, Interpreter.




Rule-Based System

## RBS components - Description

### ◇ Working Memory (WM)

- Contains facts about the world observed or derived from a rule; stored as a triplet **< object, attribute, values >**

  e.g. **< car, color, red >** : "The color of my car is red".
- Contains temporary knowledge about problem-solving session.
- Can be modified by the rules.

### ◇ Rule Base (RB)

- RB contains rules; each rule is a step in a problem solving.
- Rules are domain knowledge and modified only from outside.
- Rule syntax is **IF <condition> THEN <action>**

  e.g. **IF <(temperature, over, 20>**

  **THEN <add (ocean, swimable, yes)>**
- If the conditions are matched to the working memory and if fulfilled then rule may be fired.
- RB actions are :

  **"Add"** fact(s) to WM;

  **"Remove"** fact(s) from WM;

  **"Modify"** fact(s) in WM;

- **Interpreter**
  - It is the domain independent reasoning mechanism for RBS.
  - It selects rule from Rule Base and applies by performing action.
  - It operates on a cycle:

    **Retrieval** - Finds the rules that matches the current WM;

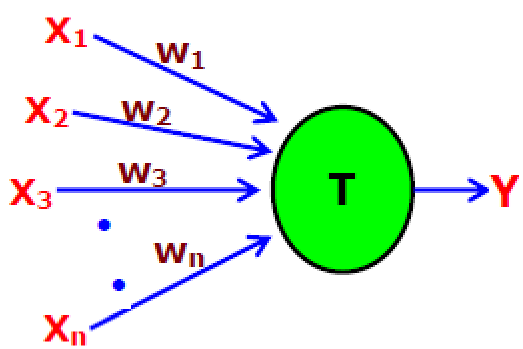
    **Refinement** - Prunes, reorders and resolves conflicts;

    **Execution** - Executes the actions of the rules in the Conflict Set, then applies the rule by performing action.

## Biology-Inspired AI Techniques

### Neural Networks (NN)

- Neural Networks model a **brain learning by example.**
- Neural networks are structures **trained** to recognize input patterns.
- Neural networks typically take a vector of input values and produce a vector of output values; inside, they train weights of "neurons".
- A Perceptron is a model of a single `trainable' neuron, shown below :




**Perceptron Model**

- $x_1, x_2, \dots, x_n$ are inputs as real numbers or boolean values depends on problem.
- $w_1, w_2, \dots, w_n$ are weights of the edges and are real valued.
- $T$ is the threshold and is real valued.
- $y$ is the output and is boolean.

If the net input which is $w_1 x_1 + w_2 x_2 + \dots + w_n x_n$ is greater than the threshold $T$ then output $y$ is **1** else **0**.

- Neural networks use **supervised learning**, in which inputs and outputs are known and the goal is to build a representation of a function that will approximate the input to output mapping.

## Genetic Algorithms (GA)

- **GAs are part of evolutionary computing**, a rapidly growing area of AI.
- **Genetic algorithms** are implemented as a computer simulation, where techniques are inspired by evolutionary biology.
- **Mechanics of biological evolution**
  - Every **organism** has a set of **rules**, describing how that organism is built, and encoded in the genes of an organism.
  - The genes are connected together into long strings called **chromosomes.**
  - Each gene represents a specific **trait** (feature) of the organism and has several different settings, e.g. setting for a hair color gene may be black or brown.
  - The genes and their settings are referred as an organism's **genotype.**
  - When two organisms mate they share their genes. The resultant **offspring** may end up having half the genes from one parent and half from the other. This process is called **cross over.**
  - A gene may be **mutated** and expressed in the organism as a completely new trait.
- Thus, Genetic Algorithms are a way of solving problems by mimicking processes, the nature uses, Selection, Crosses over, Mutation and Accepting to evolve a solution to a problem.

- **Genetic Algorithm Steps**

  (1) **[Start]** Generate random population of **n** chromosomes (Encode suitable solutions for the problem)

  (2) **[Fitness]** Evaluate the fitness **f(x)** of each chromosome **x** in the population.

  (3) **[New population]** Create a new population by repeating following steps until the new population is complete.

  - (a) **[Selection]** Select two parent chromosomes from a population according to their fitness.
  - (b) **[Crossover]** With a crossover probability, cross over the parents to form new offspring (children). If no crossover was performed, offspring is the exact copy of parents.
  - (c) **[Mutation]** With a mutation probability, mutate new offspring at each locus (position in chromosome).
  - (d) **[Accepting]** Place new offspring in the new population.

  (4) **[Replace]** Use new generated population for a further run of the algorithm.

  (5) **[Test]** If the end condition is satisfied, stop, and return the best solution in current population.

  (6) **[Loop]** Go to step 2.

- Genetic Algorithms does **unsupervised learning** – the right answer is not known beforehand.

## Reinforcement Learning (RL)

- RL is **learning from interaction with an environment;** from the consequences of action, rather than from explicit teaching.
- RL is conducted within the mathematical framework of Markov decision processes (MDPs).
- The basic RL model consists : a set of "environment states - **S** " ; a set of "actions - **A**"; and a set of scalar "rewards - **R**".
- The decision-making agent interacts with its environment so as to maximize the cumulative reward it receives over time. The steps are:
  - At each time $t$, the agent perceives its environment state $s_t$ and the set of possible actions $A(s_t)$ .
  - It chooses an action $a \in A(s_t)$ and receives from the environment the new state $s_{t+1}$ and a reward $r_{t+1}$.
  - Based on these interactions, the agent develops a policy $\pi : S \rightarrow A$ which maximizes the quantity $R = r_0 + r_1 + . . + r_n$ for MDPs.
- RL methods focus on the kind of learning and decision making problems that people face in their normal, everyday lives.

# Branches of AI

- **Logical AI**
  - Logic is a **language for reasoning**; a collection of rules used while doing logical reasoning.
  - Types of logic
    - Propositional logic - logic of sentences
    - predicate logic - logic of objects
    - logic involving uncertainties
    - Fuzzy logic - dealing with fuzziness
    - Temporal logic, etc
  - **Propositional logic** and **Predicate logic** are fundamental to all logic
    - Propositional logic
      - Propositions are "Sentences"; either true or false but not both.
      - A sentence is smallest unit in propositional logic
      - If proposition is true, then truth value is "true"; else "false"
      - Example : Sentence "Grass is green"; Truth value " true"; Proposition is "yes"

## Predicate logic

- Predicate is a function may be true or false for arguments
- Predicate logic are rules that govern quantifiers
- Predicate logic is propositional logic added with quantifiers
- Examples:

  "The car Tom is driving is blue",

  "The sky is blue",

  "The cover of this book is blue"

  **Predicate** **is blue**, give a name **B** ;

  **Sentence** represented as **B(x)**; read B(x) as "x is blue" ;

  **Object** represented as **x** .

## Search in AI

- **Search is a problem-solving technique** that systematically consider all possible action to find a path from initial state to target state.

- **Search techniques** are many; the most fundamental are
  - Depth first
  - Breadth first
  - Hill climbing
  - Least cost

- **Search components**
  - Initial state - First location
  - Available actions - Successor function : reachable states
  - Goal test - Conditions for goal satisfaction
  - Path cost - Cost of sequence from initial state to reachable state

- **Search objective**
  - Transform initial state into goal state - find a sequence of actions.

- **Search solution**
  - Path from initial state to goal - optimal if lowest cost.

## Pattern Recognition (PR)

- **Definitions** : from the literature
  - 'The assignment of a physical object or event to one of pre-specified categories' – Duda and Hart
  - 'The science that concerns the description or classification (recognition) of measurements' – Schalkoff
  - 'The process of giving names Ω to observations X ' – Schürmann
  - Pattern Recognition is concerned with answering the question 'What is this?' – Morse
  - 'A problem of estimating density functions in a high-dimensional space and dividing the space into the regions of categories or classes' – Fukunaga

- **Pattern recognition problems**
  - Machine vision - Visual inspection, ATR
  - Character recognition – Mail sorting, processing bank cheques
  - Computer aided diagnosis - Medical image/EEG/ECG signal analysis
  - Speech recognition - Human Computer Interaction, access
- **Approaches for Pattern recognition**
  - Template Matching
  - Statistical classification
  - Syntactic or Structural matching

Each of these approaches are explained in the next slide.

**Template Matching**

Match with stored template considering translation, rotation and scale changes; measure similarity (correlation) based on training set.

**Statistical classification**

Each pattern is represented in terms of **d** features (measurements) and viewed as a point in a d-dimensional space. Using training sets establish decision boundaries in the feature space - following decision theoretic or discriminant analysis approaches.

**Syntactic or Structural matching**

Complex pattern is composed of sub-patterns and the relations; they themselves are built from simpler / elementary sub-patterns are called primitives. The patterns are viewed as sentences belonging to a language, primitives are viewed as the alphabet of the language. The sentences are generated according to a grammar. A large collection of complex patterns can be described by a small number of primitives and grammatical rules. The grammar for each pattern class are inferred from the training samples.

## Neural networks

Neural networks are viewed as weighted directed *graphs* in which the *nodes* are artificial neurons and directed *edges* (with weights) are connections between neurons input-output. Neural networks have the *ability to learn* complex nonlinear input-output relationships from the sequential training procedures, and adapt themselves to input data.

## Applications requiring Pattern recognition

- Image Proc / Segmentation
- Computer Vision
- Medical Diagnosis
- Man and Machine Diagnostics
- Seismic Analysis
- Industrial Inspection
- Financial Forecast

## Knowledge Representation

- **How do we represent what we know?**

  Knowledge is a collection of facts. To manipulate these facts by a program, a suitable representation is required. A Good representation facilitates problem solving.

- **Knowledge representation formalisms (techniques)**

  Different types of knowledge require different types of representation.

  - **Predicate logic :**

    Predicate is a function may be TRUE for some arguments, and FALSE for others.

  - **Semantic networks :**

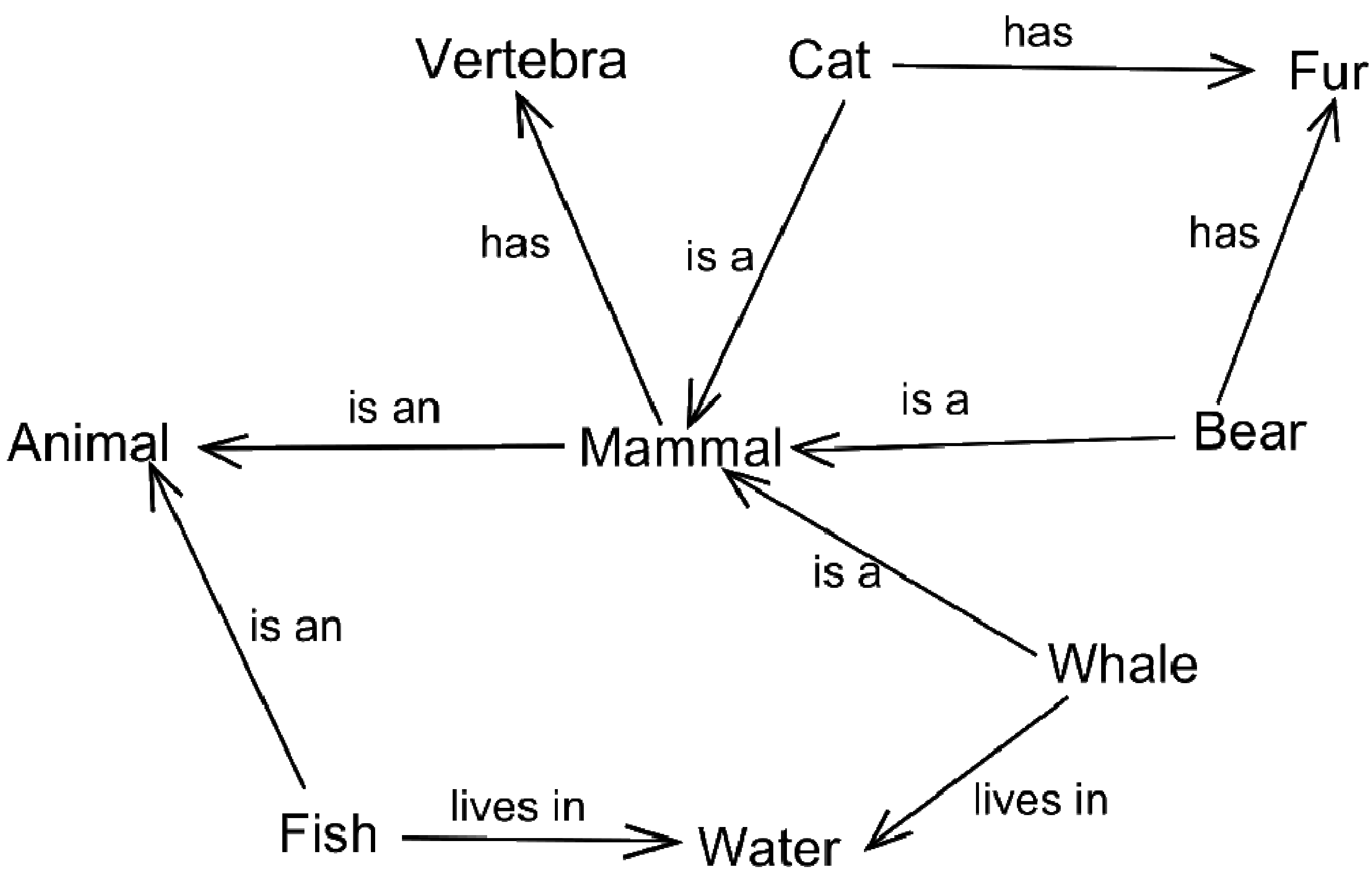
    A semantic net is just a graph, where the nodes represent concepts, and the arcs represent binary relationships between concepts.

  - **Frames and scripts :**

    A frame is a data structure that typically consists of : Frame name, Slot-filler (relations target), Pointers (links) to other Frames, Instantiation Procedure (inheritance, default, consistency). The Scripts are linked sentences using frame-like structures; e.g., a record of sequence of events for a given type of occurrence.

  - **Production rules :**

    Consists of a set of rules about behavior; a production consists two parts: a precondition (or **IF**) and an action (or **THEN**); if a production's precondition matches the current state of the world, then the production is said to be triggered.




**Semantic Net**

*CHAIR frame*

A-kind-of: furniture

number-of-legs: an integer (default=4)

style-of-back: straight, cushioned, ...

number-of-arms: 0,1,2

*John's-chair frame*

a-kind-of: furniture

number-of-legs: 4

style-of-back: cushioned

number-of-arms: 0

**Frames and Scripts**

## Rules

These are also called **condition-action rules**.

These components of a rule-based system have the form:

if <condition> then <conclusion>

or

if <condition> then <action>

*Example:*

**if** patient has high levels of the enzyme ferritin in their blood
**and** patient has the Cys282→Tyr mutation in HFE gene
**then** conclude patient has haemochromatosis*

* medical validity of this rule is not asserted here

Rules can be evaluated by:

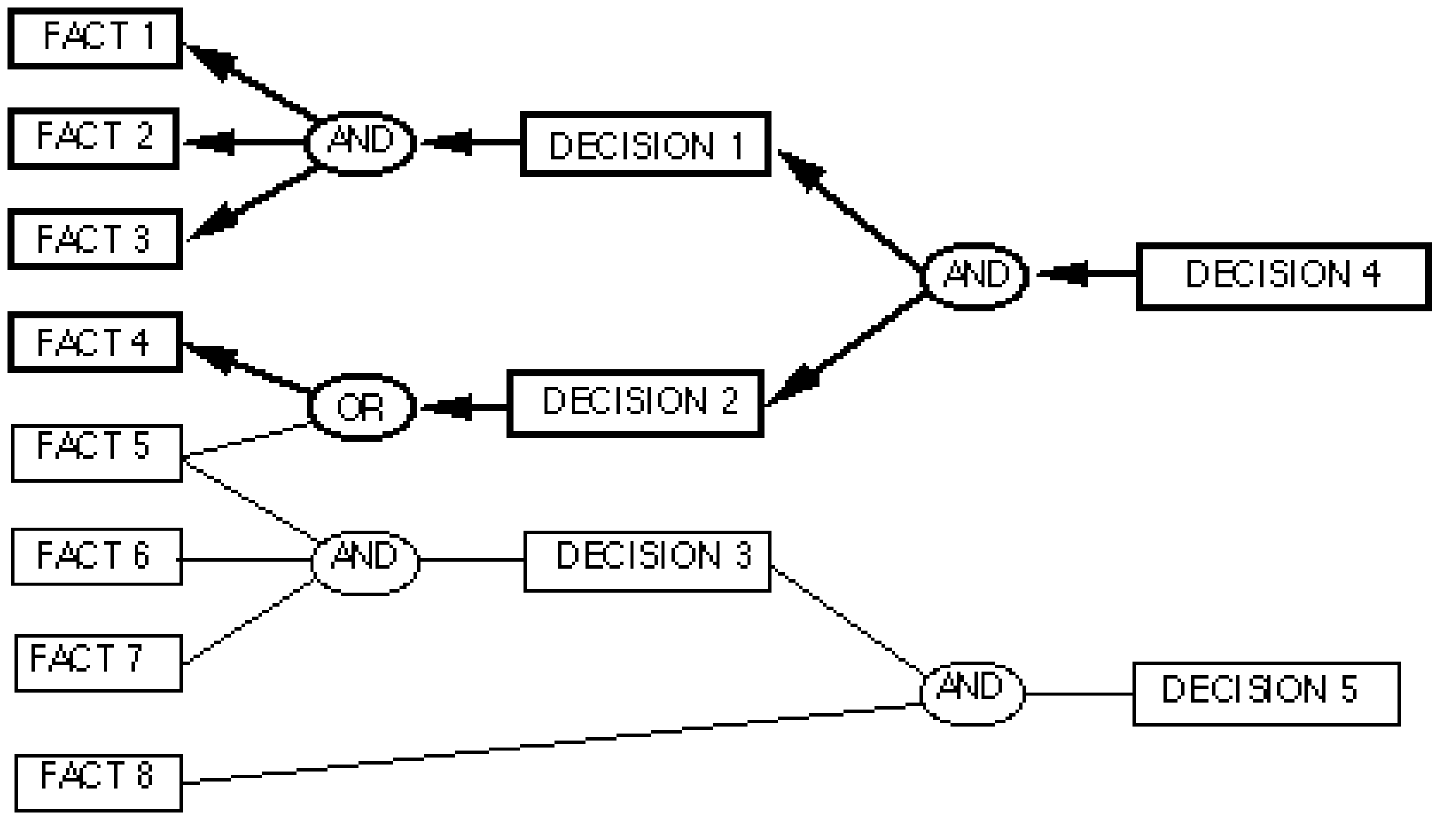
•backward chaining

•forward chaining

**Backward Chaining**

To determine if a decision should be made, work backwards looking for justifications for the decision.

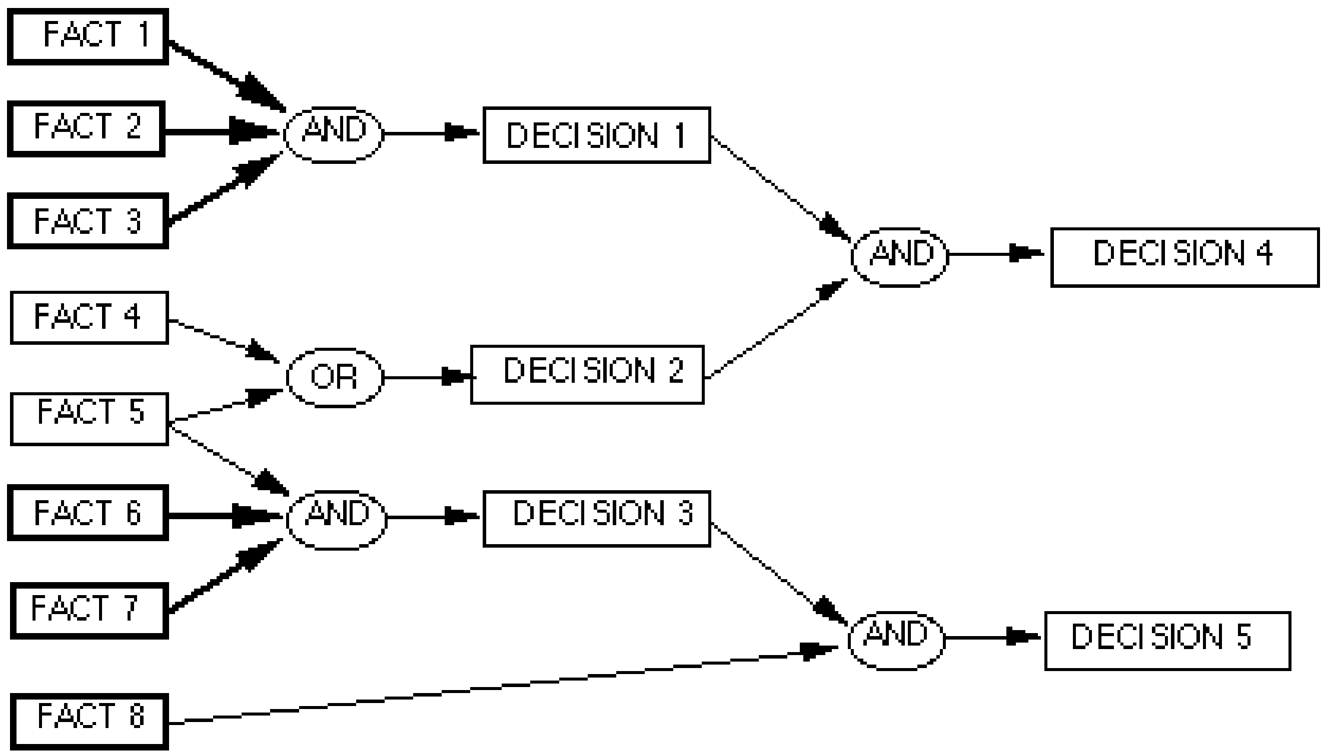
Eventually, a decision must be justified by facts.



**Forward Chaining**

Given some facts, work forward through inference net.

Discovers what conclusions can be derived from data.



## Production Rules

A production rule system consists of

- a set of rules
- working memory that stores temporary data
- a forward chaining inference engine

## Match-Resolve-Act Cycle

The match-resolve-act cycle is what the inference engine does.

**loop**

  match conditions of rules with contents of working memory

  **If** no rule matches **then** stop

  resolve conflicts

act (i.e. perform conclusion part of rule)**end loop**

**BAGGER**

- Bagger is a simple rule-based system that describes how to pack items at a supermarket check-out. It has been attributed to Patrick Winston of MIT.
- While explaining Bagger, we shall describe a number of potential strategies for conflict resolution.
- Bagger's working memory has an associated table of attributes of the objects (stock items) at the supermarket.
- There are 4 steps in Bagger, and Bagger uses a Working Memory item called "Step" to keep track of where it is up to.
- Each rule checks the value of "Step" as part of its **if** part, and will be applicable only to one of the four steps.
- This makes it easier to be sure that the rules will not interact in unexpected ways (a pitfall in creating rule-based systems).

**Steps in Bagger**

1. **Check order**: Check what the customer has selected; look to see if something is missing, suggest additions.
2. **Pack large items**: Put the large items in the bag; put big bottles first.
3. **Pack medium items**: Put in the medium sized items; put frozen food in plastic bags.
4. **Pack small items**: Put in the small items wherever there is room.

More details: http://www.cse.unsw.edu.au/~billw/cs9414/notes/kr/rules/rules.html

## Inference

Inference is the act or process of deriving a conclusion based solely on what one already knows; it is deduction of new facts from old ones; Logic captures inference.

- **Deductive Inference**
  - It is never false; inference is true if premise is true.
  - A traditional logic is based on deduction; it is a method of exact inference; there is no possibility of mistake if rules are followed exactly.
  - The information required is complete, precise, and consistent .
  - A logic is monotonic, if the truth of a proposition does not change when new information are added to the system.

- **Inductive Inference**
  - It may be correct or incorrect inference, because in real world the information is incomplete, inexact, inconsistent;
  - A logic is inductive, also called Induction or Inductive reasoning, if the process of reasoning in which the premises of an argument are believed to support the conclusion but do not ensure it.
  - A logic is non-monotonic, if the truth of a proposition may change when a new information (axioms) is added to or an old information is deleted from the system.
  - The reasoner draws conclusions tentatively reserving the right to retract them in the light of further information.
  - Example: When we hear of a bird, we human infer that bird can fly, but this conclusion can be reversed when we hear that it is a penguin; the bird penguin cannot fly.

## Common Sense Knowledge and Reasoning

Common sense is the mental skills that most people have.

- It is the ability to analyze a situation based on its context.
  - **People can think** because the brain contain vast libraries of common sense knowledge and has means for organizing, acquiring, and using such knowledge.
  - **Computer can not think;** the computers programs do many things, they can play chess at the level of best players but cannot match capabilities of a 3 year old child at recognizing objects. Currently, computers lack common sense.

- Researchers have divided common sense capability into :
  - Common sense knowledge and
  - Common sense reasoning.

- Teaching computers common sense
  - Project "**OpenMind**" at MIT - Here the goal is to teach a computer things that human take them for granted; here the knowledge is represented in the form of Semantic net, Probabilistic graphical models, and Story scripts.
  - Project "**Cyc**" – It is an attempt to manually build a database of human common sense knowledge; it has 1.5 million collection of common sense facts, but still far away from several hundred million needed.

## Learning

Programs learn from what the facts or the behaviors can represent.

- **Definitions**
  - Herbert Simon 1983 – "Learning denotes changes in the system that are adaptive in the sense that they enable the system to do the same task or tasks more efficiently and more effectively the next time."
  - Marvin Minsky 1986 – "Learning is making useful changes in the working of our mind."
  - Ryszard Michalski 1986 – "Learning is constructing or modifying representations of what is being experienced."
  - Mitchell 1997 – "A computer program is said to learn from experience **E** with respect to some class of tasks **T** and performance measure **P**, if its performance at tasks in **T**, as measured by **P**, improves with experience **E**."

## Major Paradigms of Machine Learning

- **Rote : Learning by memorization;** Saving knowledge so that it can be used again.
- **Induction : Learning by example;** Process of learning by example where a system tries to induce a general rule from a set of observed instances.
- **Analogy : Learning from similarities;** Recognize similarities in information already stored; can determine correspondence between two different representations.
- **Genetic Algorithms : Learning by mimicking processes nature uses;** Part of evolutionary computing, a way of solving problems by mimicking processes, nature uses, *selection, crosses over, mutation and accepting* to evolve a solution to a problem.
- **Reinforcement : Learning from actions;** Assign rewards, +ve or -ve; at the end of a sequence of steps, it learns which actions are good or bad.

# Some questions to answer

1. What are Constraint Satisfaction Problems?
2. What is supervised versus unsupervised learning?
3. What are neural networks and how do they relate to AI?
4. What is the difference between machine learning and deep learning?
5. Where is AI being used already?
6. What can AI do – and what is it not able to do?
7. Do AI and data protection go together?

## Planning

A plan is a representation of a course of action.

Planning is a problem solving technique.

Planning is a reasonable series of actions to accomplish a goal.

- **Planning programs**

  Start with facts about the world, particularly

  - facts about the effects of actions,
  - facts about the particular situation, and
  - statement of a goal.

- **Benefits of planning**

  - reducing search,
  - resolving goal conflicts, and
  - providing a basis for error recovery.

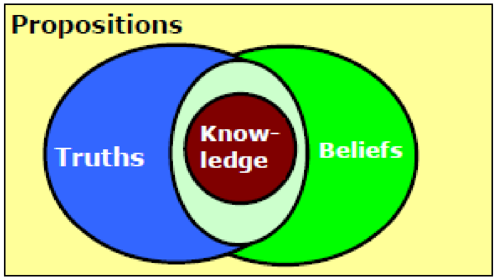
- **Strategy for planning**

  A strategy is just a sequence of actions. From facts the program generate a strategy for achieving the goal.

## Epistemology

Epistemology is the theory of knowledge.

- There are various kinds of knowledge:
  - knowing how to do something (e.g., how to ride a bicycle),
  - knowing someone in person, and
  - knowing a place or a city.
- Epistemology is the study of knowledge and justified belief.




  - Consider a knowledge of proposition - a schema ***'S knows that P'*** ;
  - Question asked - what are the necessary and sufficient conditions?

    **S** knows that **P** if and only if

    **P** is true;

    **S** believes that **P** is true; and

    **S** is justified in believing that **P** is true.

  *Gettier* claimed that the above account of knowledge is insufficient.

- Epistemology is the study of the kinds of knowledge that are required for solving problems in the world.

## Ontology

Ontology is concerned with existence; a study of the categories of things that exist or may exist in some domain.

- Ontology is a data model, represents a domain and is used to reason about the objects in that domain and the relations between them.
- Ontology is used in artificial intelligence, as a form of knowledge representation about the world or some part of it.
- Ontology generally describe:
  - **Individuals** (instances): the basic or ground level objects
  - **Classes:** sets, collections, or types of objects.
  - **Attributes:** properties, features, characteristics, or parameters that objects can have and share.
  - **Relations:** ways the objects can be related to one another.
- Ontology is a specification of a conceptualization.

## Heuristics

Heuristics are simple, efficient rules;

- Heuristics are in common use as **Rule of thumb;**

  In computer science, a heuristic is an algorithm with provably good run times and with provably good or optimal solution.

- Heuristics are intended to gain computational performance or conceptual simplicity, potentially at the cost of accuracy or precision.

- People use heuristics to make decisions, come to judgments, and solve problems, when facing complex problems or incomplete information. These rules work well under most circumstances.

- In AI programs, the heuristic functions are :

  - used to measure how far a node is from goal state.

  - used to compare two nodes, find if one is better than the other.

## Genetic programming (GP)

Genetic programming is an automated method for creating program from a high-level problem statement.

- GP starts from a high-level statement of the requirements of a problem and attempts to produce a computer program that solves the problem.
- The user (human) communicates the high-level statement of the problem to the GP system by performing certain well-defined preparatory steps.
- The major five preparatory steps, the human user require to specify for the GP to be evolved are :
  - ◇ Set of terminals (variables , functions, and constants)
  - ◇ Set of primitive functions
  - ◇ Fitness measure (for fitness of individuals in the population),
  - ◇ Controlling parameters for the run, and
  - ◇ Termination criterion for designating the result of the run.
- The "Run" of genetic programming executes a series of well-defined problem-independent steps (the flowchart).

# Applications of AI

**Game playing**

- Games are Interactive computer program, an emerging area in which the goals of human-level AI are pursued.
- Games are made by creating human level artificially intelligent entities, e.g. enemies, partners, and support characters that act just like humans.
- Game play is a search problem defined by:
  - Initial state - board
  - Expand function - build all successor states
  - Cost function - payoff of the state
  - Goal test - ultimate state with maximal payoff

# Applications of AI Contd..

- Game playing is characterized by:
  - "Unpredictable" opponent
  - Need to specify move for every possible opponent reply.
  - Time limits - games become boring if there is no action for too long a time; opponents are unlikely to find goal, must approximate.

- Computer Games
  - Computers perform at champion level games, examples : Checkers, Chess, Othello, Backgammon.
  - Computers perform well games, example : Bridge
  - Computers still do badly, example : Go, Hex

- The **Deep Blue** Chess program won over world champion Gary Kasparov.

## Speech Recognition

- A process of converting a speech signal to a sequence of words;
- In 1990s, computer speech recognition reached a practical level for limited purposes.
- Using computers recognizing speech is quite convenient, but most users find the keyboard and the mouse still more convenient.
- The typical usages are :
  - Voice dialing (Call home),
  - Call routing (collect call),
  - Data entry (credit card number).
  - Speaker recognition.
- The spoken language interface PEGASUS in the American Airlines' EAASY SABRE reservation system, allows users to obtain flight information and make reservations over the telephone.

## Understanding Natural Language

Natural language processing (NLP) does automated generation and understanding of natural human languages.

- **Natural language generation system**
  Converts information from computer databases into normal-sounding human language

- **Natural language understanding system**
  Converts samples of human language into more formal representations that are easier for computer programs to manipulate.

- **Some major tasks in NLP**
  - Text-to-Speech (TTS) system :

    converts normal language text into speech.
  - Speech recognition (SR) system :

    process of converting a speech signal to a sequence of words;
  - Machine translation (MT) system :

    translate text or speech from one natural language to another.
  - Information retrieval (IR) system :

    search for information from databases such as Internet or World Wide Web or Intranets.

## Computer Vision

- It is a combination of concepts, techniques and ideas from : Digital Image Processing, Pattern Recognition, Artificial Intelligence and Computer Graphics.
- The world is composed of 3-D objects, but the inputs to the human eye and computers' TV cameras are 2-D.
- Some useful programs can work solely in 2-D, but full computer vision requires partial 3-D information that is not just a set of 2-D views.
- At present there are only limited ways of representing 3-D information directly, and they are not as good as what humans evidently use.

- **Examples**
  - **Face recognition :**

    the programs in use by banks
  - **Autonomous driving :**

    The ALVINN system, autonomously drove a van from Washington, D.C. to San Diego, averaging 63 mph day and night, and in all weather conditions.
  - **Other usages**

    Handwriting recognition, Baggage inspection, Manufacturing inspection, Photo interpretation, etc .

## Expert Systems

Systems in which human expertise is held in the form of rules

- It enable the system to diagnose situations without the human expert being present.

- **A Man-machine system** with specialized problem-solving expertise.
  The "expertise" consists of knowledge about a particular domain, understanding of problems within that domain, and "skill" at solving some of these problems.

- **Knowledge base**
  A knowledge engineer interviews experts in a certain domain and tries to embody their knowledge in a computer program for carrying out some task.

- One of the **first expert systems was MYCIN in 1974**, which diagnosed bacterial infections of the blood and suggested treatments.

**Searching among rules, many of which can apply to a situation**:

If-conditions

*the infection is primary-bacteremia*
*AND the site of the culture is one of the sterile sites*
*AND the suspected portal of entry is the gastrointestinal tract*

*THEN*

*there is suggestive evidence (0.7) that infection is bacteroid*

(from MYCIN)

Expert systems rely on knowledge of human experts, e.g.

- **Diagnosis and Troubleshooting :**
  deduces faults and suggest corrective actions for a malfunctioning device or process
- **Planning and Scheduling :**
  analyzing a set of goals to determine and ordering a set of actions taking into account the constraints; e.g. airline scheduling of flights
- **Financial Decision Making :**
  advisory programs assists bankers to make loans, Insurance companies to assess the risk presented by the customer, etc.
- **Process Monitoring and Control :**
  analyzes real-time data, noticing anomalies, predicting trends, and controlling optimality and do failure correction.

# References


1. "Artificial Intelligence", by Elaine Rich and Kevin Knight, (2006), McGraw Hill companies Inc., Chapter 1-22, page 1-613.
2. "Artificial Intelligence: A Modern Approach" by Stuart Russell and Peter Norvig, (2002), Prentice Hall, Chapter 1-27, page 1-1057.
3. "Computational Intelligence: A Logical Approach", by David Poole, Alan Mackworth, and Randy Goebel, (1998), Oxford University Press, Chapter 1-12, page 1-608.
4. "Artificial Intelligence: Structures and Strategies for Complex Problem Solving", by George F. Luger, (2002), Addison-Wesley, Chapter 1- 16, page 1-743.
5. "AI: A New Synthesis", by Nils J. Nilsson, (1998), Morgan Kaufmann Inc., Chapter 1-25, Page 1-493.
6. "Artificial Intelligence: Theory and Practice", by Thomas Dean, (1994), Addison-Wesley, Chapter 1-10, Page 1-650.
7. Related documents from open source, mainly internet. An exhaustive list is being prepared for inclusion at a later date.