RV College of Engineering®

Go, change the world

Unit-1

Artificial Intelligence and Machine learning

21AI52

Outline

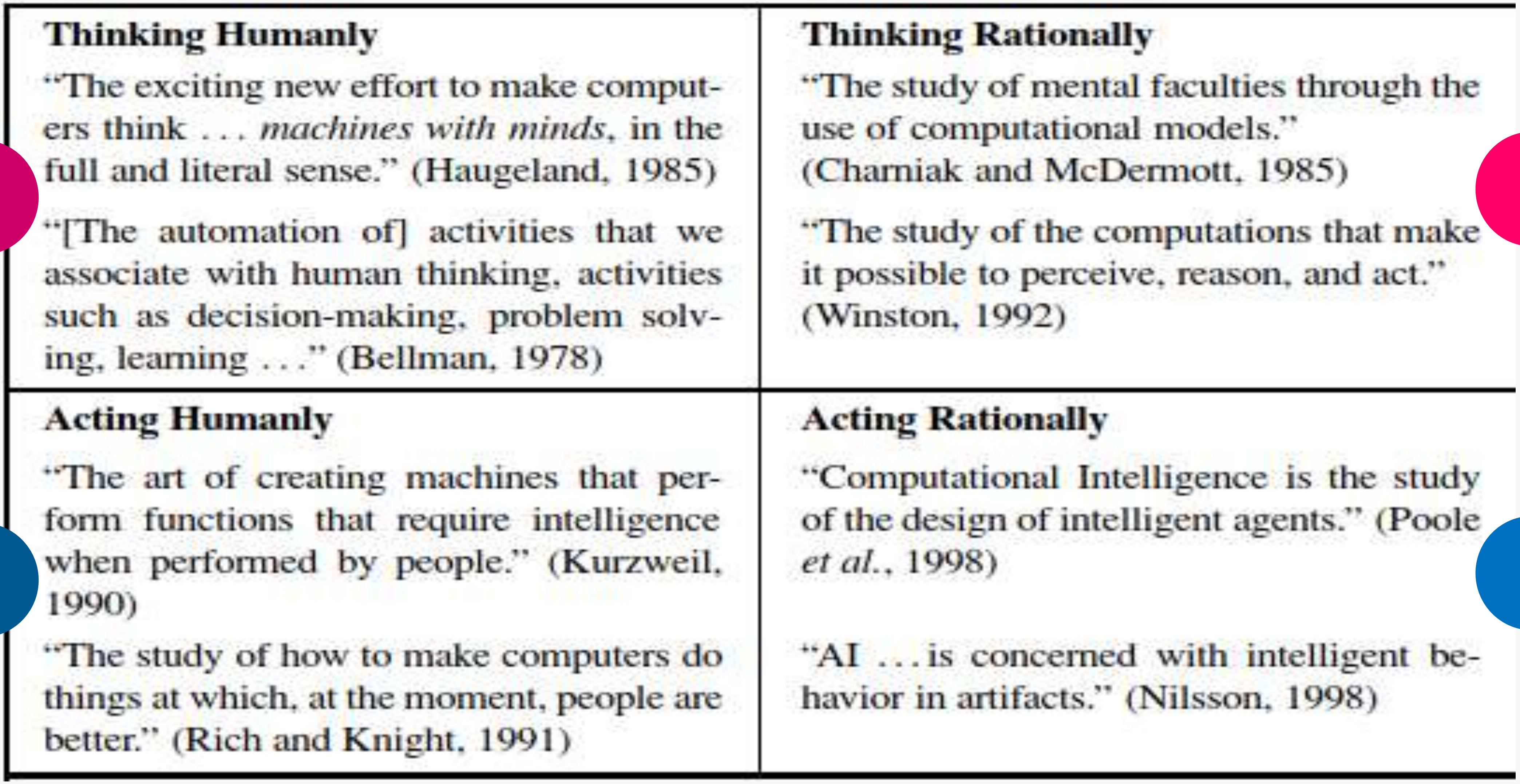
- Introduction
  - → What is AI ?
  - → Foundation of Artificial Intelligence
  - → History of Artificial Intelligence
  - → The State of the Art
- Intelligent Agents
  - → Introduction
  - → How Agents should Act
  - → Structure of Intelligent Agents
- Problem Solving
  - → Solving the Problems by Searching
  - → Search Strategies
  - → Avoiding the Repeated States

Introduction

What is AI ?

- Artificial Intelligence (AI) takes the advantages of the computers, and machines to mimic the problem solving, and decision making capabilities of the human beings
  - Computer: Intelligence enough to process and analyse the data
  - Machine: Contains no intelligence, waiting for the instructions from the user or computer
- The definitions vary in two main dimensions
  - First: Concerned with thought processes and reasoning
  - Second: Concerned with the behavior
- Remember: AI system is rational if it does the right thing

What is AI ? (Contd.)



1 **Thinking Humanly** "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)	2 **Thinking Rationally** "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)
3 **Acting Humanly** "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)	4 **Acting Rationally** "Computational Intelligence is the study of the design of intelligent agents." (Poole *et al.*, 1998) "AI . . . is concerned with intelligent behavior in artifacts." (Nilsson, 1998)

What is AI ? (Contd.)

- Definition of AI are categorized into following four categories

1. System that thinks like Humans	2. Systems that think Rationally
3. System that act like Humans	4. Systems that act Rationally

- A Human-Centered Approach is an empirical science, involving hypothesis and experimental
- A Rationalist Approach involves a combination of mathematics and engineering.

Rationalist Approach = Mathematics + Engineering

- Generally, all four approaches have been followed, during the development of AI systems

What is AI ? (Contd.)

Acting Humanly: The Turing Test Approach

- The Turing Test, proposed by Alan Turing (1950), was designed to provide a satisfactory operational definition of intelligence.
- Turing Machine defined to exhibit intelligent behavior as the ability to achieve human-level performance in all cognitive tasks, sufficient to fool an interrogator.
- Roughly speaking, the test he proposed is that the computer should be interrogated by a human via a teletype, and passes the test if the interrogator cannot tell if there is a computer or a human at the other end.
- The Turing Machine (Computer) would need to possess the following Human capabilities

What is AI ? (Contd.)

Acting Humanly: The Turing Test Approach

- Natural Language Processing (NLP): to enable it to communicate successfully in English (or some other human language)
- Knowledge Representation: to store information provided before or during the interrogation;
- Automated Reasoning to use the stored information to answer questions and to draw new conclusions;
- Machine Learning (ML) to adapt to new circumstances and to detect and extrapolate patterns

▪ Turing's test deliberately avoided direct physical interaction between the interrogator and the computer, because physical simulation of a person is unnecessary for intelligence

▪ However, total Turing Test includes a video signal so that the interrogator can test the subject's perceptual abilities, as well as the opportunity for the interrogator to pass physical objects "through the hatch.

What is AI ? (Contd.)

Acting Humanly: The Turing Test Approach

- To pass the total Turing Test, the computer will need
  - Computer Vision to perceive objects, and
  - Robotics to manipulate the objects (move arround)
- Acting like a human comes up primarily when AI programs interact with people, as when an expert system explains how it came to its diagnosis, or a natural language processing system has a dialogue with a user.
- These programs must behave according to certain normal conventions of human interaction in order to make themselves understood.

What is AI ? (Contd.)

Thinking Humanly: The Cognitive Modelling Approach

- If the program is said to be thinking like human, some ways to determine the human thinking need to be listed
- There are two ways to understand the actual working of the human minds
  - Through Introspection: Trying to catch the human thoughts as they go by
  - Through Psychological Experiments
- Once we have a sufficiently precise theory of the mind, it becomes possible to express the theory as a computer program.
- If the program's input/output and timing behavior matches human behavior, that is evidence that some of the program's mechanisms may also be operating in humans.

What is AI ? (Contd.)

Thinking Humanly: The Cognitive Modelling Approach

- The interdisciplinary field of ***Cognitive Science*** brings together ***Computer Models from AI*** and experimental techniques from ***Psychology*** to try to construct precise and testable theories of the workings of the human mind.
- Real Cognitive Science, is based on experimental investigation of actual humans or animals
- AI and Cognitive Science continue to fertilize each other, especially in the areas of vision, natural language, and learning

What is AI ? (Contd.)

⇒Thinking Rationally: The laws of thought Approach

- Logicians in the 19th century developed a precise notation for statements about all kinds of objects in the world and the relations among them.
- By 1965, programs existed that could, given enough time and memory, take a description of a problem in logical notation and find the solution to the problem, if one exists
- Logicist tradition within artificial intelligence hopes to build on such programs to create intelligent systems.
- There are two main obstacles to this approach
  - It is not easy to take informal knowledge and state it in the formal terms required by logical notation
  - There is a big difference between being able to solve a problem and doing so in practice
- Even problems with just a few dozen facts can exhaust the computational resources of any computer unless it has some guidance as to which reasoning steps to try first
- Although both of these obstacles apply to any attempt to build computational reasoning systems, they appeared first in the Logicist tradition because the power of the representation and reasoning systems are well-defined and fairly well understood

What is AI ? (Contd.)

Acting rationally: The rational agent approach

- An Agent is just something that perceives and acts
- Acting Rationally: Doing right things
- In this approach, AI is viewed as the study and construction of rational agents
- In the *"Laws of Thought"* approach to AI, the whole emphasis was on correct inferences.
- Making correct inferences is sometimes part of being a rational agent, because one way to act rationally is to reason logically to the conclusion that a given action will achieve one's goals, and then to act on that conclusion.

What is AI ? (Contd.)

Acting rationally: The Rational Agent Approach

- On the other hand, correct inference is not all of rationality, because there are often situations where there is no provably correct thing to do, yet something must still be done
- There are also ways of acting rationally that cannot be reasonably said to involve inference. For example, pulling one's hand off of a hot stove is a reflex action that is more successful than a slower action taken after careful deliberation
- To allow Rational Actions, all of the “Cognitive Skills" required for the Turing Test are present.
- Thus, we need the ability to represent knowledge and reason with it because this enables us to reach good decisions in a wide variety of situations.

What is AI ? (Contd.)

Acting rationally: The Rational Agent Approach

- One important point to keep in mind: Achieving perfect rationality—always doing the right thing—is not possible in complicated environments. The computational demands are just too high.
- Limited Rationality- Acting appropriately when there is not enough time to do all the computations one might like.

The Foundation of AI

- This section provide a brief history of AI
- Although AI itself is a young field, it has inherited many ideas, viewpoints, and techniques from other disciplines.
  - From over 2000 years of tradition in philosophy, theories of reasoning and learning have emerged, along with the viewpoint that the mind is constituted by the operation of a physical system.
  - From over 400 years of mathematics, which comprises of formal theories of logic, probability, decision making, and computation.
  - From psychology, the tools to investigate the human mind, and a scientific language to express the resulting theories. From linguistics, the theories of the structure and meaning of language.
  - Finally, from computer science, the tools with which to make AI a reality.

The Foundation of AI (Contd.)

Foundation of AI

Philosophy

1. Can formal rules be used to draw valid conclusions ?
2. Where does knowledge come from ?
3. How does the knowledge lead to action

Mathematics

1. What are the rules to draw the valid Conclusions ?
2. What can be computed ?
3. How do we reason with uncertain information ?

Neuroscience

1. How do brain process the information ?

Psychology

1. How do brain process the information ?

Economics

1. How to we make decisions so as to maximize the payoff?
2. How should we do this when others may not go along ?

Computer Engineering

1. How can we build an Efficient Computer ?

Linguistics

1. How does language relate to thought ?

Different fields have contributed to AI, in terms of Ideas, Viewpoints, and Techniques

Intelligent Agents

Introduction

- An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors.
- A Human Agent has eyes, ears, and other organs for sensors, and hands, legs, mouth, and other body parts for effectors
- A Robotic Agent substitutes cameras and infrared range finders for the sensors and various motors for the effectors.
- A generic agent is diagrammed in shown in below Figure.
- Examples for Agents
  - Human Agent
  - Robotic Agent

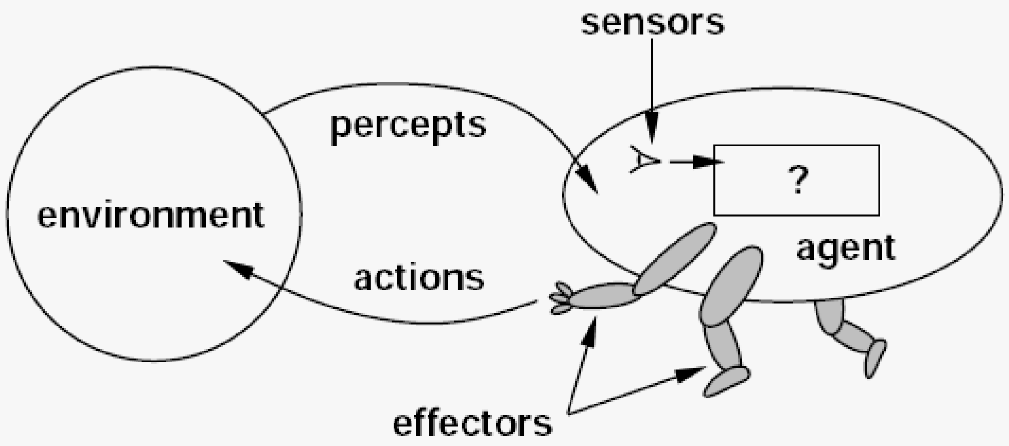


Figure. Agents interact with Environments through Sensors and Effectors

How Agent should Act

Rational Agent

- A rational agent is one that does the right thing
- As a first approximation, it can be say that the right action is the one that will cause the agent to be most successful.
- The *Performance Measure* of the Agent is based on *How and When*
- There are no single fixed measure suitable for all agents
- *Example 1*: consider the case of an agent that is supposed to vacuum a dirty floor. A probable performance measure would be the amount of dirt cleaned up in a single eight-hour shift. A more sophisticated performance measure would factor in the amount of electricity consumed and the amount of noise generated as well

How Agent should Act (Contd.)

Rational Agent

- In summary, what is rational at any given time depends on four things:
  - The performance measure that defines degree of success.
  - Everything that the agent has perceived so far. We will call this complete perceptual history the percept sequence.
  - What the agent knows about the environment.
  - The actions that the agent can perform.
- Rationality is not same as the perfection.
- Rationality maximizes the expected outcome, while perfection maximizes the actual performance
- Vacuum Cleaner Example
  - A simple agent that cleans a square if it is dirty and moves to the other square if not

How Agent should Act (Contd.)

Rational Agent

- Assumptions
  - Performance measure: 1 point for each clean square at each time step
  - Environment: is known a priori
  - Actions = {left, right, suck, no-op}
  - Agent is able to perceive the location and dirt in that location

How Agent should Act (Contd.)

⇒ Omniscience, Learning, and Autonomy

- An Omniscient Agent knows the actual outcome of its actions, and can act accordingly; but omniscience is impossible in reality.
- Rationality does not require Omniscience
- Agents can perform actions in order to modify future percepts so as to obtain useful information (Information Gathering, Exploration)
  - Information Gathering: Collecting the data from the surroundings
  - Exploration: Analyzing the data to draw the future patterns
- An agent can also *Learn* from what it perceives
- An agent is *Autonomous* if its behavior is determined by its own experience (with ability to learn and adapt)

How Agent should Act (Contd.)

⇒ Specifying the Task Environment

- Specifying the Task Environment is the first step in AI, and it is based on *PEAS* (Performance, Environment, Actuators, Sensors)
- PEAS Description of the Task Environment for an Automated taxi

Agent Name	Performance Measure	Environment	Actuators	Sensors
Taxi Driver	Safe, Fast, Legal, Comfortable Trip, Maximize profits	Roads, other traffic, pedestrians, customers	Steering, accelerator, brake, signal, horn, display	Camera, sonar, speedometer, GPS, odometer, engine sensors, keyboard

How Agent should Act (Contd.)

⇒ Specifying the Task Environment

Agent Name	Performance Measure	Environment	Actuators	Sensors
Medical Diagnosis System	Healthy patient, minimize costs, lawsuits	Patient, hospital, Staff	Display questions, tests, diagnosis, treatments, referrals	Keyboard entry of symptoms, findings, patient's answers
Satellite Image Analysis System	Correct Image Categorization	Downlink from Orbiting Satellite	Display of the Images	Color Pixel Arrays
Part-Picking robot	Percentage of Parts in correct bins	Conveyor belt with parts; bin	Jointed arm, and hand	Camera, Joint angle sensors.

How Agent should Act (Contd.)

⇒ Specifying the Task Environment

Agent Name	Performance Measure	Environment	Actuators	Sensors
Refinery Control	Purity, Yield, Safety	Refinery, Operators	Valves, Pumps, Heaters, Display	Temperature, Pressure, Chemical Sensors
Interactive English Tutor	Student's score on test	Set of Students, Testing agency	Display of exercises, suggestions, corrections	Keyboard entry

How Agent should Act (Contd.)

⇒ Properties of Task Environment

- **Fully observable Vs. Partially observable**
  - An agent's perceives the all the relevant information from the environment
  - If the agent does not have any sensors, then it is unobservable
- **Deterministic Vs. Stochastic**
  - Next state of the env. determined by current state and the agent's action
  - If the environment is deterministic except for the actions of other agents, then the environment is strategic
- **Episodic Vs. Sequential**
  - Agent's experience is divided into atomic "episodes"
  - Choice of action in each episode depends only on the episode itself

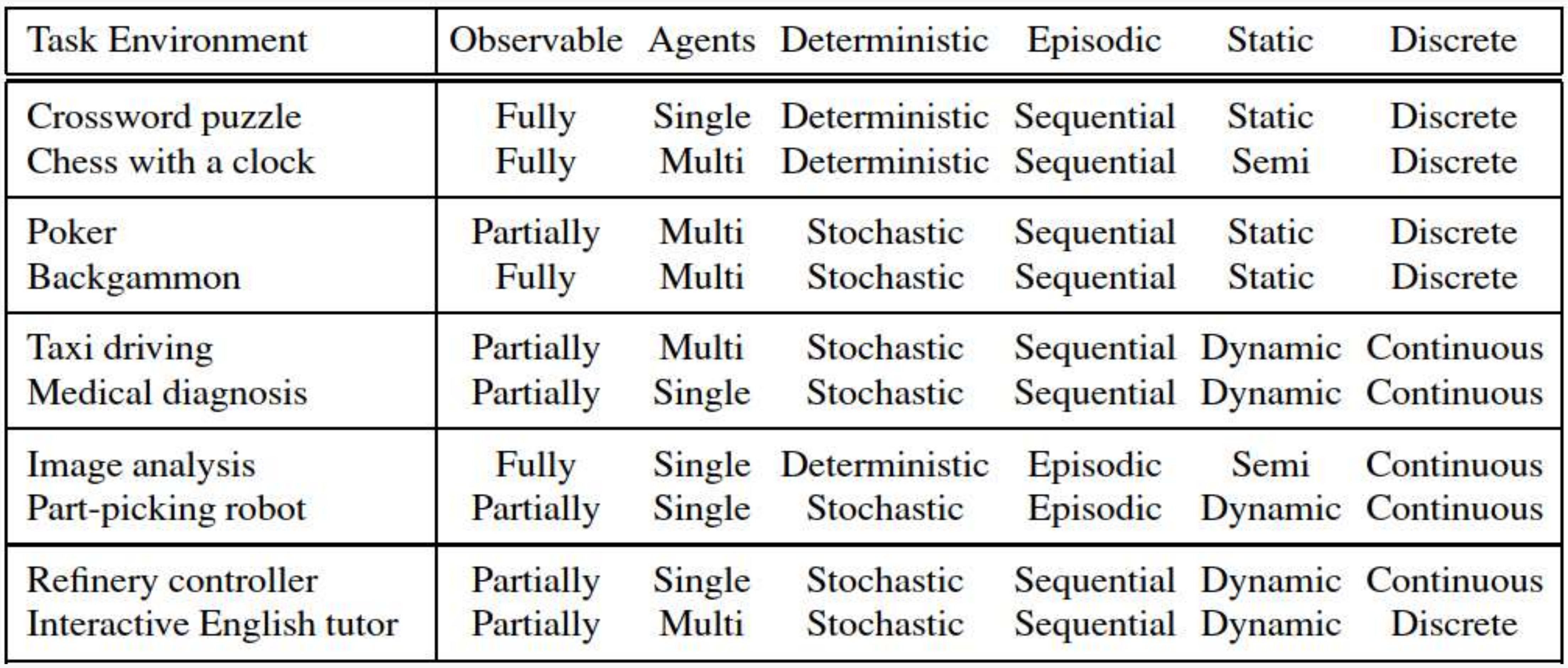
How Agent should Act (Contd.)

⇒ Properties of Task Environment

- **Static Vs. Dynamic**
  - The environment is unchanged while an agent is deliberating
  - Semi-dynamic if the environment itself doesn't change with time but the agent's performance score does
- **Discrete Vs. Continuous**
  - A limited number of distinct, clearly defined percepts and actions
- **Single agent Vs. Multiagent**
  - An agent operating by itself in an environment
- **Competitive Vs. Cooperative**
  - Chess is a competition
  - Other hand, In autonomous driving avoiding the collision, and maximizing the performance. This is Cooperative environment.

How Agent should Act (Contd.)

⇒ Properties of Task Environment



Task Environment	Observable	Agents	Deterministic	Episodic	Static	Discrete
Crossword puzzle	Fully	Single	Deterministic	Sequential	Static	Discrete
Chess with a clock	Fully	Multi	Deterministic	Sequential	Semi	Discrete
Poker	Partially	Multi	Stochastic	Sequential	Static	Discrete
Backgammon	Fully	Multi	Stochastic	Sequential	Static	Discrete
Taxi driving	Partially	Multi	Stochastic	Sequential	Dynamic	Continuous
Medical diagnosis	Partially	Single	Stochastic	Sequential	Dynamic	Continuous
Image analysis	Fully	Single	Deterministic	Episodic	Semi	Continuous
Part-picking robot	Partially	Single	Stochastic	Episodic	Dynamic	Continuous
Refinery controller	Partially	Single	Stochastic	Sequential	Dynamic	Continuous
Interactive English tutor	Partially	Multi	Stochastic	Sequential	Dynamic	Discrete

Examples of Task Environments and their Characteristics

Structure of the Agent

- The job of AI is to design the agent program that implements the agent function mapping percepts to actions
- The relationship among agents, architectures, and programs can be summed up as follows

Agent = Architecture + Program

- The architecture might be a plain computer, or it might include special-purpose hardware for certain tasks, such as processing camera images or filtering audio input
- It might also include software that provides a degree of insulation between the raw computer and the agent program, so that we can program at a higher level.
- In general, the architecture makes the percepts from the sensors available to the program, runs the program, and feeds the program's action choices to the effectors as they are generated.

Structure of the Agent (Contd.)

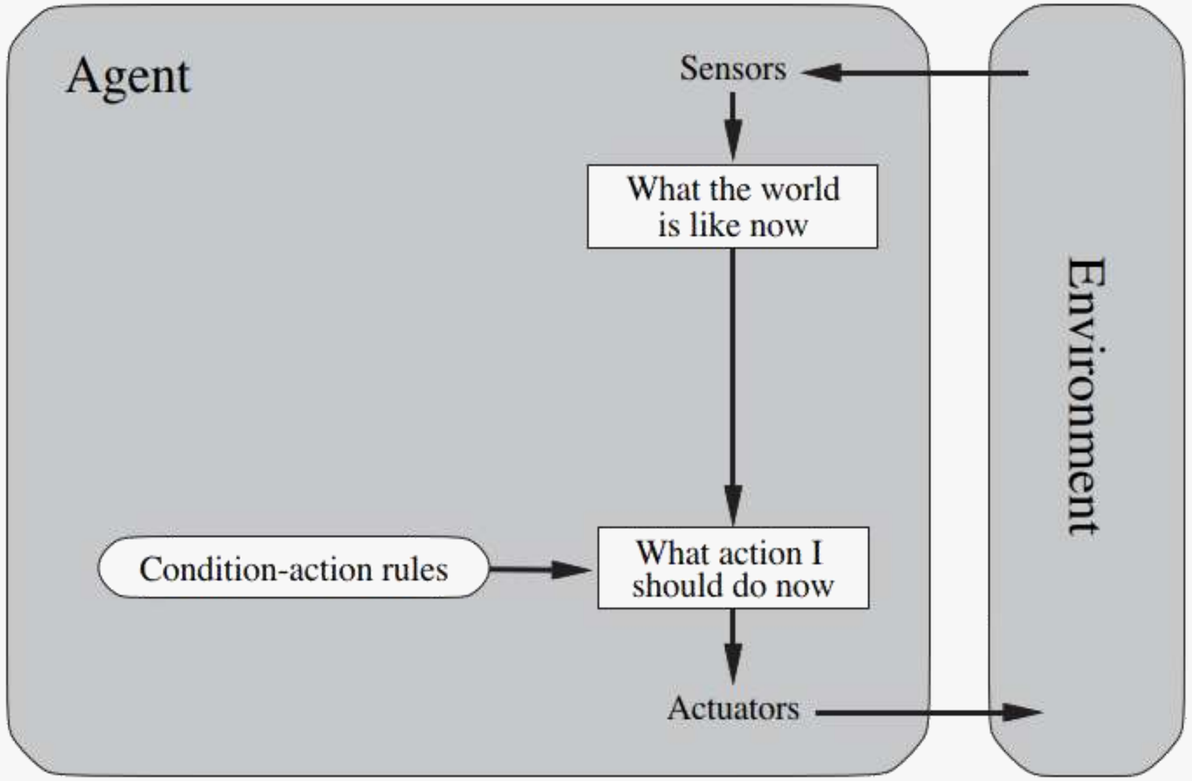
⇒ Five Basic Agent Types

- Simple reflex agents
- Model-based reflex agents
- Goal-based agents
- Utility-based agents and
- Learning agents

Structure of the Agent (Contd.)

⇒ Simple Reflex Agents

- The simplest kind of agent
- These agents select actions on the basis of the current percept, ignoring the rest of the percept history
- Simple reflex agents are, naturally, simple, but they turn out to be of limited intelligence.
- The agent will only work if the correct decision can be made on the basis of only the current percept (so only if the environment is fully observable)



```
function Reflex-Vacuum-Agent([location,status]) returns an action
if status = Dirty then return Suck
else if location = A then return Right
else if location = B then return Left
```

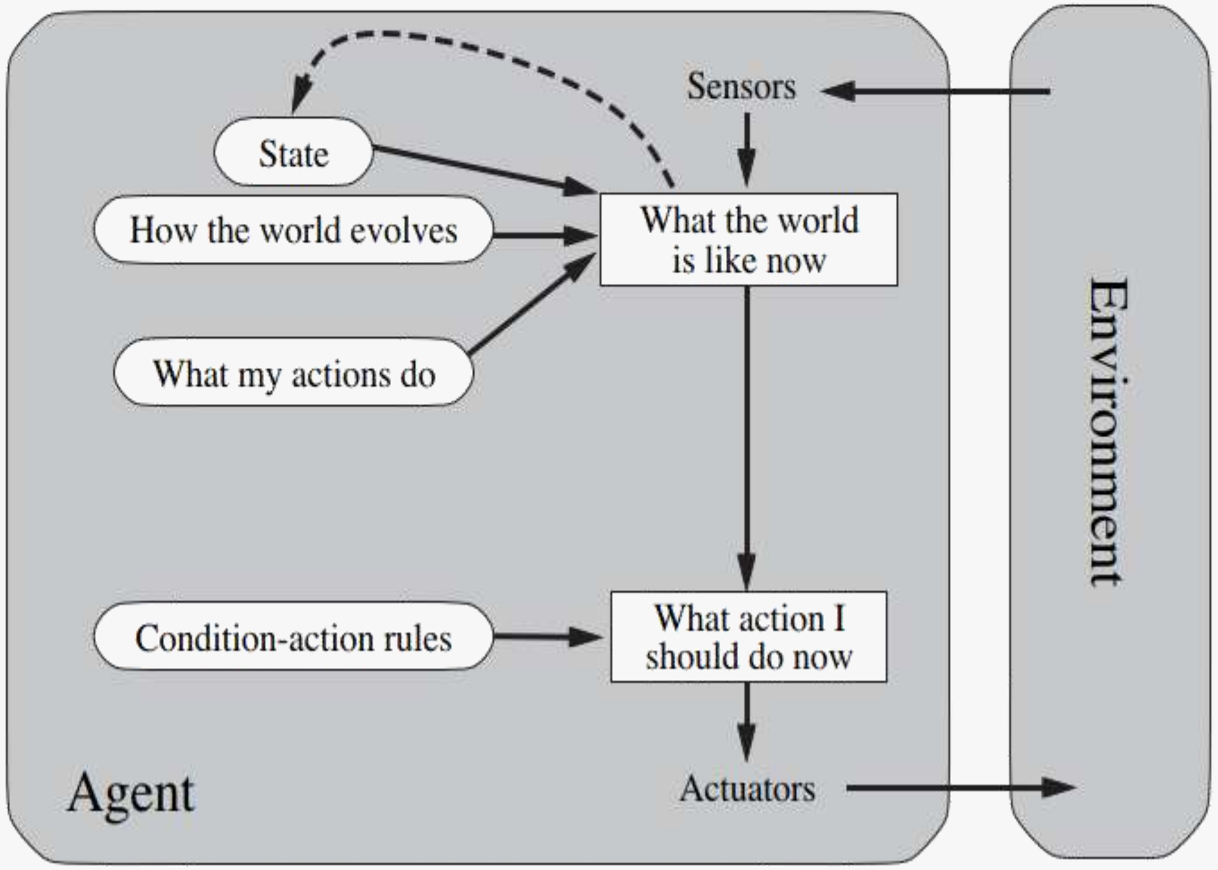
R V College of Engineering®

Structure of the Agent (Contd.)

Go, change the world

⇒ Model-based Reflex Agents

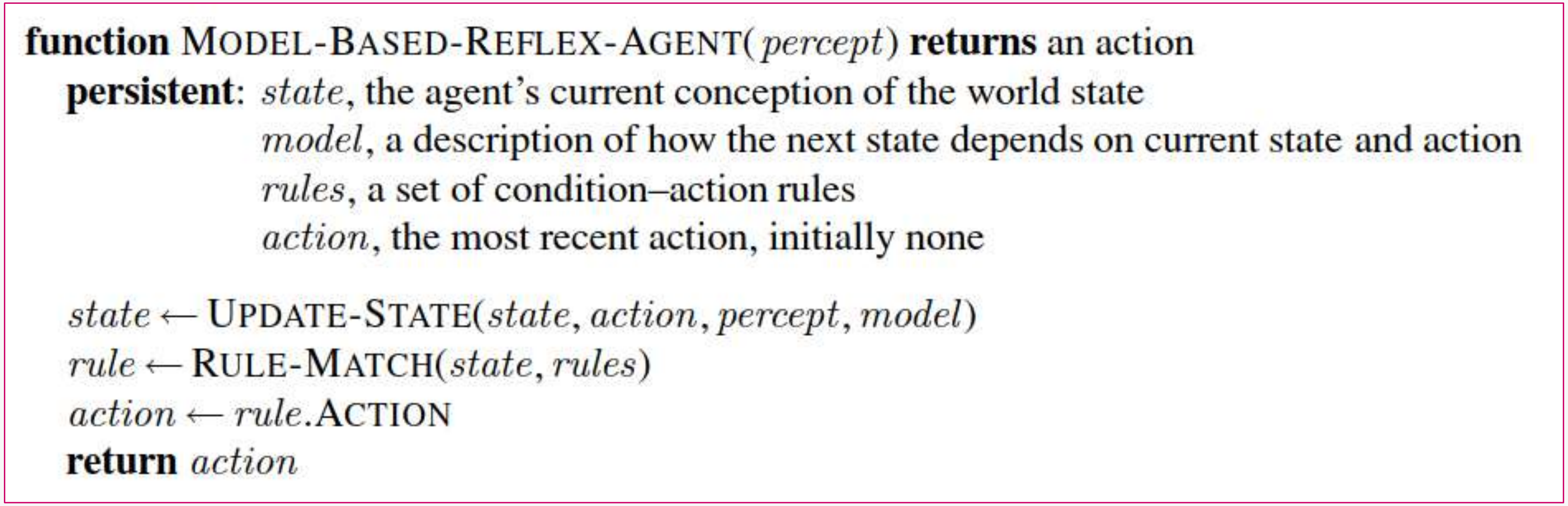
- The Model-based agent can work in a partially observable environment, and track the situation
- A model-based agent has two important factors:
  - **Model:** It is knowledge about "how things happen in the world," so it is called a Model-based agent.
  - **Internal State:** It is a representation of the current state based on percept history



- These agents have the model, "which is knowledge of the world" and based on the model they perform actions

Structure of the Agent (Contd.)

⇒ Model-based Reflex Agents



```
function MODEL-BASED-REFLEX-AGENT(percept) returns an action
    persistent: state, the agent's current conception of the world state
                model, a description of how the next state depends on current state and action
                rules, a set of condition–action rules
                action, the most recent action, initially none

    state ← UPDATE-STATE(state, action, percept, model)
    rule ← RULE-MATCH(state, rules)
    action ← rule.ACTION
    return action
```

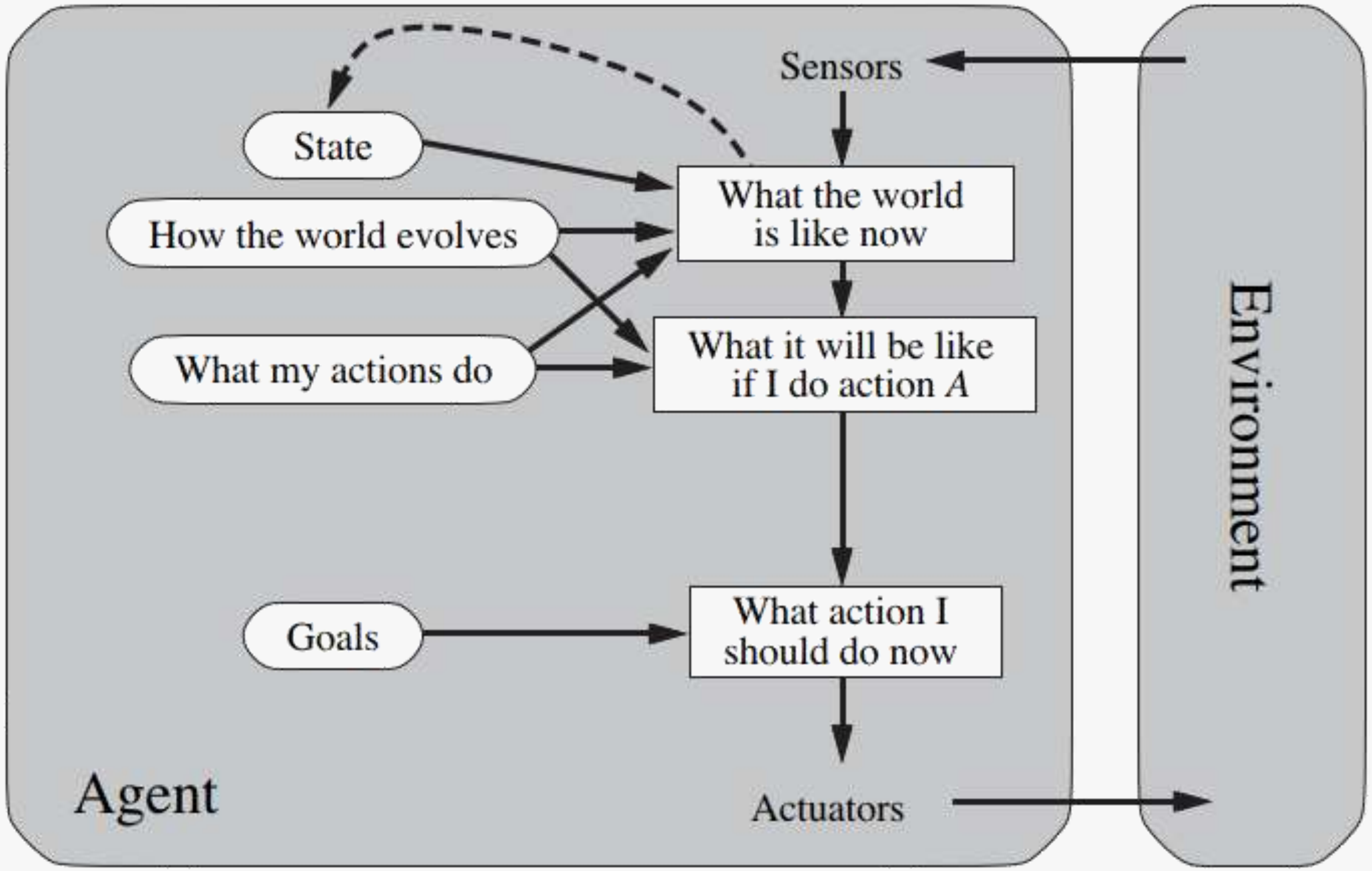
Structure of the Agent (Contd.)

⇒ Goal-based Reflex Agents

- The knowledge of the current state environment is not always sufficient to decide for an agent to what to do.
- The agent needs to know its goal which describes desirable situations.
- Goal-based agents expand the capabilities of the model-based agent by having the "goal" information
- They choose an action, so that they can achieve the goal.
- These agents may have to consider a long sequence of possible actions before deciding whether the goal is achieved or not.
- Such considerations of different scenario are called searching and planning, which makes an agent proactive.

Structure of the Agent (Contd.)

⇒ Goal-based Reflex Agents


Sensors
State
What the world is like now
How the world evolves
What my actions do
What it will be like if I do action A
Goals
What action I should do now
Agent
Actuators
Environment

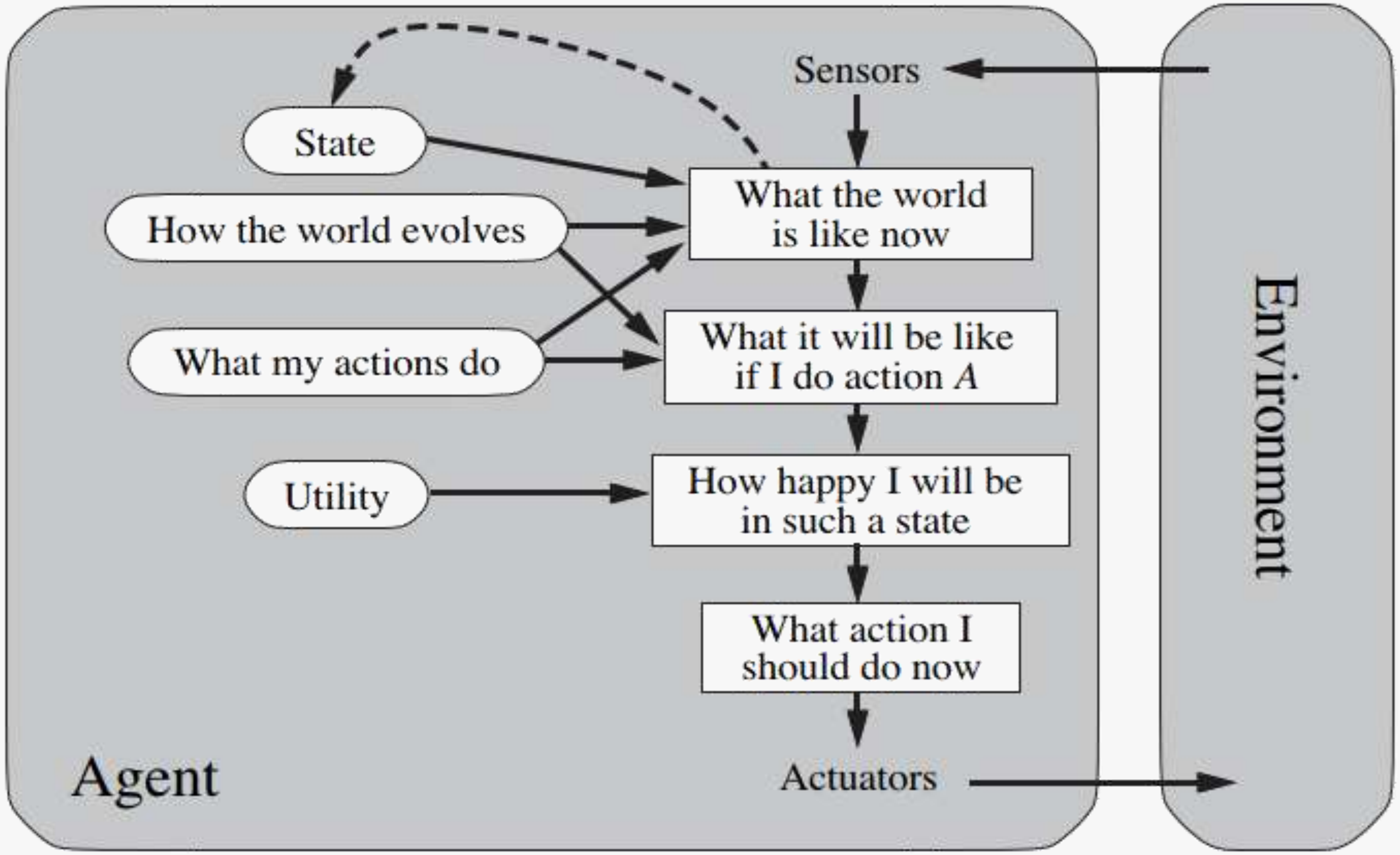
Structure of the Agent (Contd.)

⇒ Utility-based agents

- These agents are similar to the goal-based agent but provide an extra component of utility measurement which makes them different by providing a measure of success at a given state
- Utility-based agent act based not only goals but also the best way to achieve the goal.
- The Utility-based agent is useful when there are multiple possible alternatives, and an agent has to choose in order to perform the best action.

Structure of the Agent (Contd.)

⇒ Utility-based agents


Sensors
State
How the world evolves
What my actions do
Utility
What the world is like now
What it will be like if I do action *A*
How happy I will be in such a state
What action I should do now
Agent
Actuators
Environment

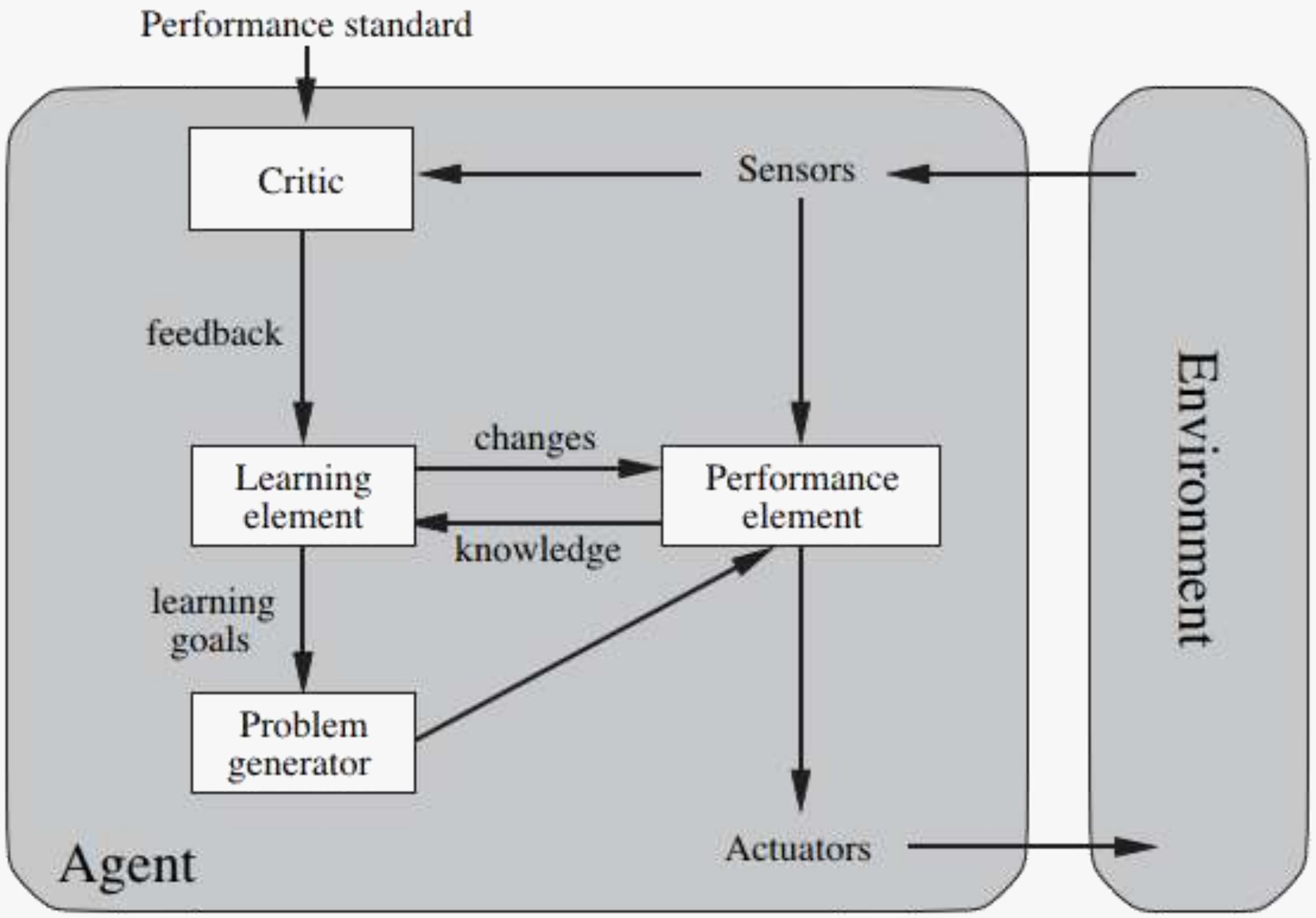
Structure of the Agent (Contd.)

⇒ Learning Agents

- A learning agent in AI is the type of agent which can learn from its past experiences, or it has learning capabilities
- It starts to act with basic knowledge and then able to act and adapt automatically through learning
- A learning agent has mainly four conceptual components, which are:
  - Learning Element: It is responsible for making improvements by learning from environment
  - Critic: Learning element takes feedback from critic which describes that how well the agent is doing with respect to a fixed performance standard.
  - Performance Element: It is responsible for selecting external action
  - Problem Generator: This component is responsible for suggesting actions that will lead to new and informative experiences.
- Hence, learning agents are able to learn, analyze performance, and look for new ways to improve the performance

Structure of the Agent (Contd.)

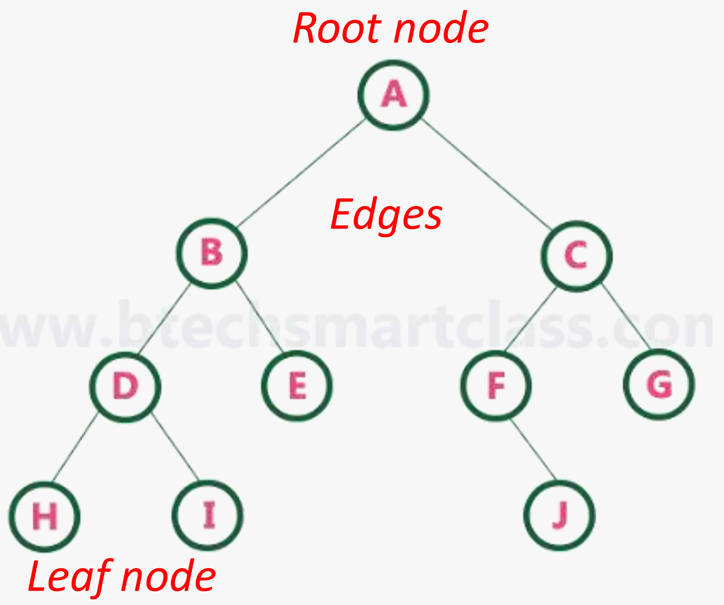
⇒ Learning Agents



Problem Solving

Problem-Solving Agents

- In Artificial Intelligence, Search techniques are universal problem-solving methods
- ***Rational agents*** *or* ***Problem-solving agents*** in AI mostly used these search strategies or algorithms to solve a specific problem and provide the best result.
- Problem-solving agents are the *Goal-based agents*
- Terminologies used in Search Algorithms
  - ***Search:*** Searching is a step by step procedure to solve a search-problem
  - ***Search Tree:*** A tree representation of search problem is called *Search tree*. The root of the search tree is the *root node* which is corresponding to the *Initial state*
  - ***Path Cost:*** It is a function which assigns a numeric cost to each path
  - ***Solution:*** It is an action sequence which leads from the start node to the goal node
  - ***Optimal Solution:*** If a solution has the lowest cost among all solution



www.btechsmartclass.com

Search Strategies (Contd.)

⇒ Properties of Search Algorithms

- *Completeness:* A search algorithm is said to be complete if it guarantees to return a solution if at least any solution exists for any random input.
- *Optimality:* If a solution found for an algorithm is guaranteed to be the best solution (lowest path cost) among all other solutions, then such a solution for is said to be an optimal solution.
- *Time Complexity:* Time complexity is a measure of time for an algorithm to complete its task.
- *Space Complexity:* It is the maximum storage space required at any point during the search, as the complexity of the problem.

Search Strategies

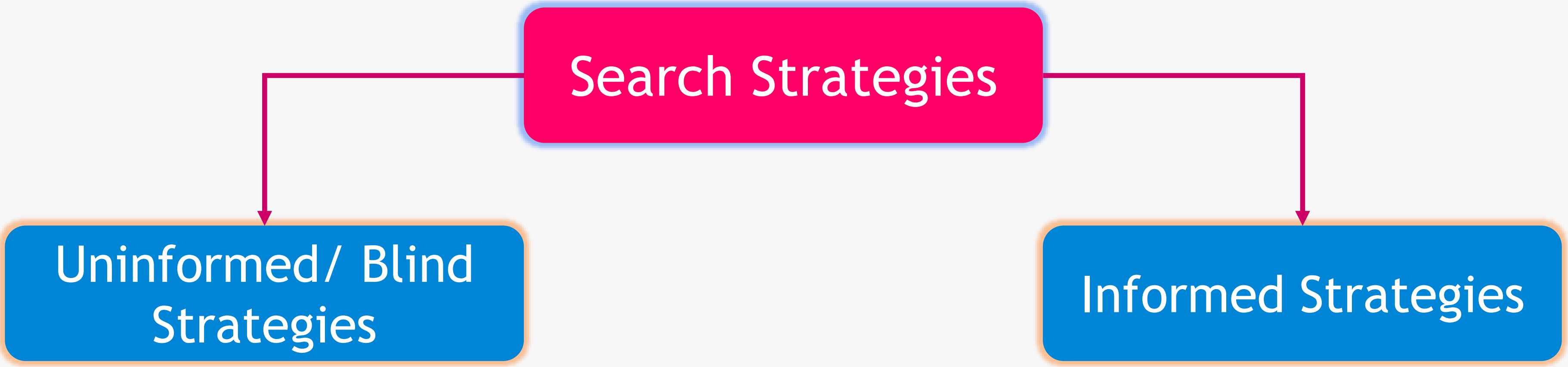
⇒ Uniformed Search

- The uninformed search does not contain any domain knowledge such as closeness, the location of the goal. It operates in a brute-force way as it only includes information about how to traverse the tree and how to identify leaf and goal nodes.
- It is also called *Blind search*. It examines each node of the tree until it achieves the goal node.

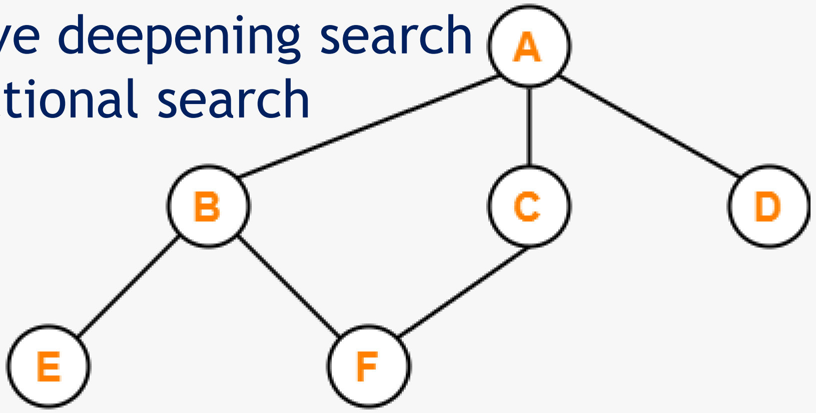
⇒Informed Search

- Informed search algorithms use domain knowledge. In an informed search, problem information is available which can guide the search.
- Informed search strategies can find a solution more efficiently than an uninformed search strategy.
- Informed search is also called a *Heuristic Search*

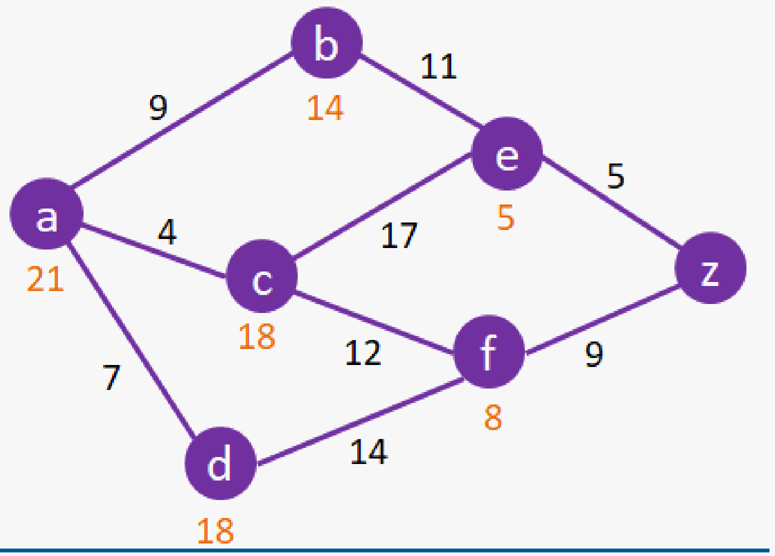
Search Strategies (Contd.)



1. Breadth First Search
2. Uniform Cost Search
3. Depth First Search
4. Depth-limited Search
5. Iterative deepening search
6. Bidirectional search



1. Breadth First Search
2. A* Search



Uninformed Search (Contd.)

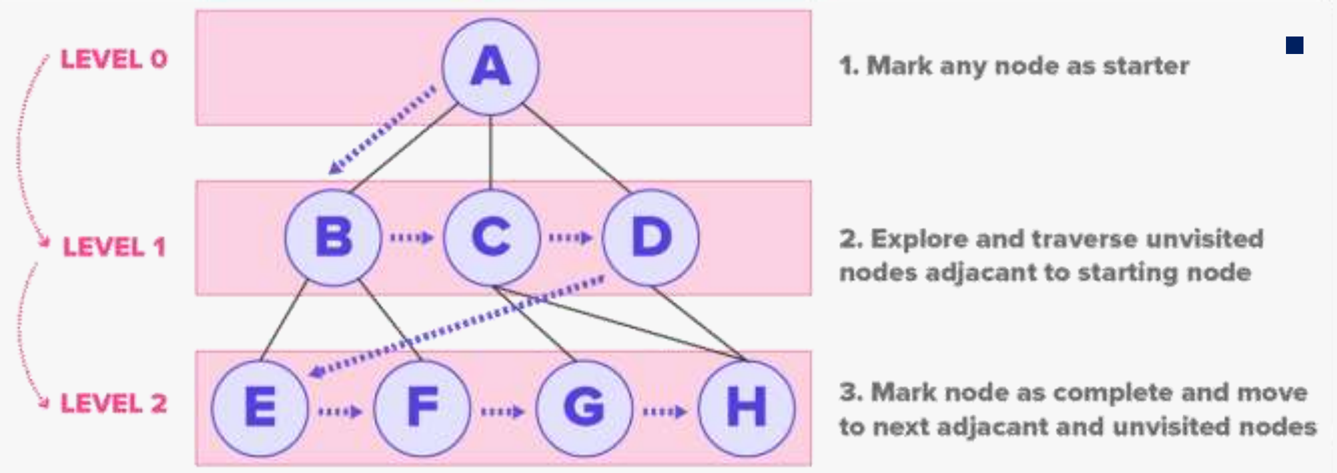
- Breadth First Search (BFS) is the most common search strategy for traversing a tree or graph.
- This algorithm searches breadthwise in a tree or graph, so it is called breadth-first search.
- BFS algorithm starts searching from the root node of the tree and expands all successor node at the current level before moving to nodes of next level.
- Breadth-first search implemented using FIFO queue data structure.
- Advantage
  - If there are more than one solutions for a given problem, then BFS will provide the minimal solution which requires the least number of steps.
- Dis-Advantages
  - It requires lots of memory since each level of the tree must be saved into memory to expand the next level.
  - BFS needs lots of time if the solution is far away from the root node

R V College of Engineering®

Uninformed Search (Contd.)

Go, change the world

- In BFS, root node is expanded first, then all the successors of the root node are expanded next, then their successors, and so on.
- Breadth-first search can be implemented by calling TREE-SEARCH with an empty fringe that is a first-in-first-out (FIFO) queue, assuring that the nodes that are visited first will be expanded first. In other words, calling TREE-SEARCH(*Problem*, FIFO-QUEUE()) results in a breadth-first search.


LEVEL 0
LEVEL 1
LEVEL 2
A
B
C
D
E
F
G
H
1. Mark any node as starter
2. Explore and traverse unvisited nodes adjacent to starting node
3. Mark node as complete and move to next adjacent and unvisited nodes

- The FIFO queue puts all newly generated successors at the end of the queue, which means that shallow nodes are expanded before deeper nodes

```
function BREADTH-FIRST-SEARCH(problem) returns a solution or failure
  return GENERAL-SEARCH(problem,ENQUEUE-AT-END)
```

Uninformed Search (Contd.)

⇒ Depth First Search

- Depth-first search is a recursive algorithm for traversing a tree or graph data structure
- Depth-first search starts from the root node and follows each path to its greatest depth node before moving to the next path
- DFS uses a stack data structure for its implementation
- Advantage
  - DFS requires very less memory as it only needs to store a stack of the nodes on the path from root node to the current node.
  - It takes less time to reach to the goal node than BFS algorithm (if it traverses in the right path).
- Disadvantage
  - There is the possibility that many states keep re-occurring, and there is no guarantee of finding the solution.
  - DFS algorithm goes for deep down searching and sometime it may go to the infinite loop

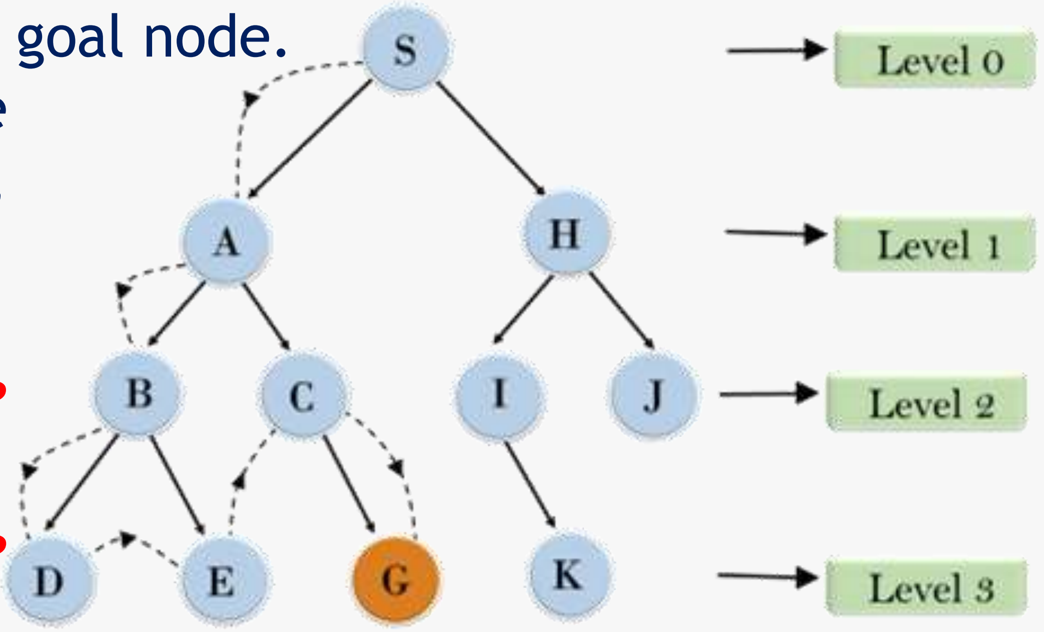
Uninformed Search (Contd.)

- It will start searching from root node S, and traverse A, then B, then D and E, after traversing E, it will backtrack the tree as E has no other successor and still goal node is not found. After backtracking it will traverse node C and then G, and here it will terminate as it found goal node.
- In the below search tree, we have shown the flow of depth-first search, and it will follow the order as:

Root Node → Left Node → Right Node

(or)

Root Node → Right Node → Left Node

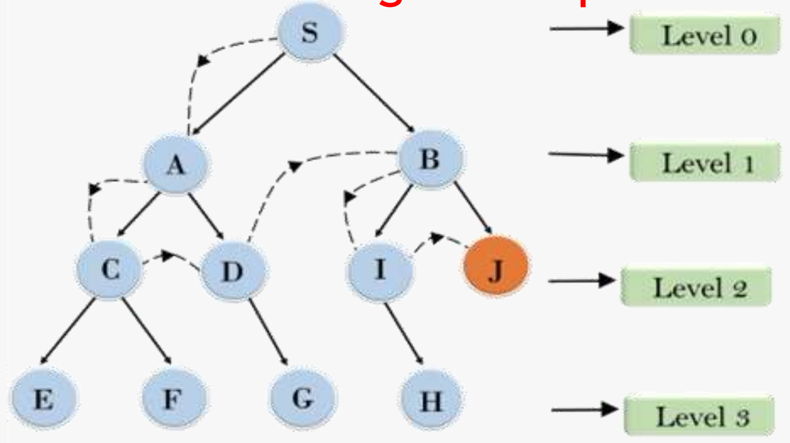


```
function DEPTH-FIRST-SEARCH(problem) returns a solution, or failure
    GENERAL-SEARCH(problem,ENQUEUE-AT-FRONT)
```

Uninformed Search (Contd.)

⇒ Depth-limited Search

- A depth-limited search algorithm is similar to depth-first search with a predetermined limit. Depth-limited search can solve the drawback of the infinite path in the Depth-first search
- In this algorithm, the node at the depth limit will treat as it has no successor nodes further
- Depth-limited search can be terminated with two Conditions of failure
  - Standard failure value: It indicates that problem does not have any solution
  - Cutoff failure value: It defines no solution for the problem within a given depth limit
- Advantages
  - Depth-limited search is Memory efficient
- Disadvantages
  - It may not be optimal if the problem has more than one solution



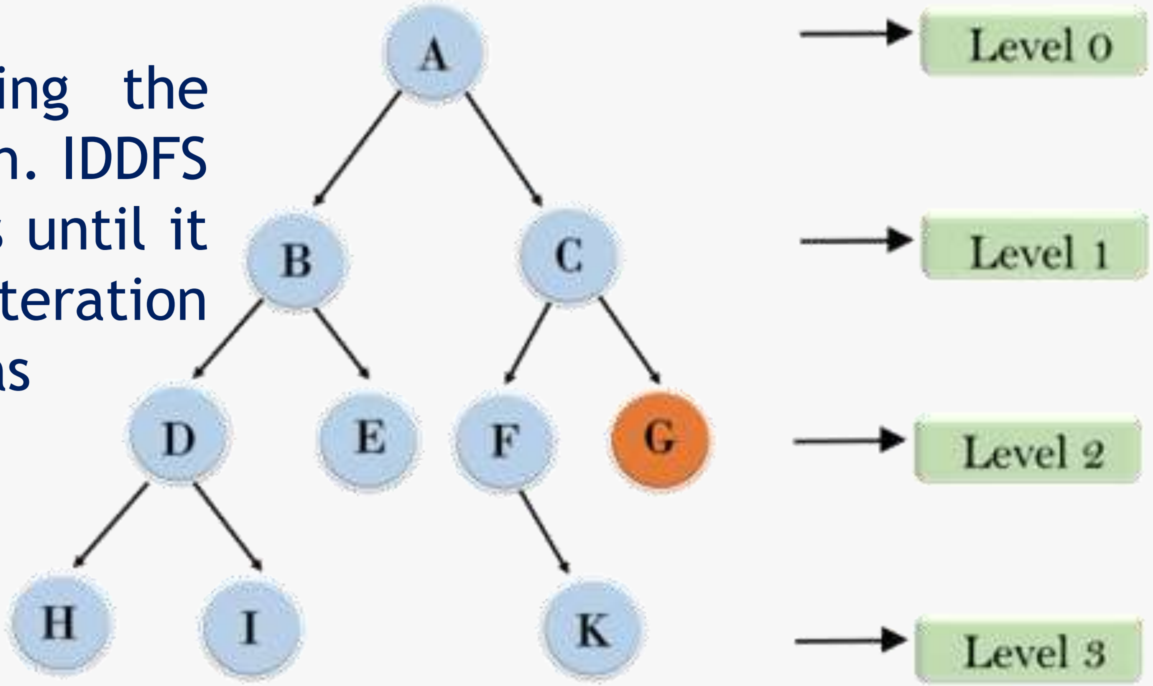
Uninformed Search (Contd.)

⇒Iterative Deepening Depth-First Search

- The iterative deepening algorithm is a *combination of DFS and BFS* algorithms
- This search algorithm finds out the best depth limit and does it by gradually increasing the limit until a goal is found
- This algorithm performs depth-first search up to a certain *“Depth Limit"*, and it keeps increasing the depth limit after each iteration until the goal node is found
- This Search algorithm combines the benefits of Breadth-first search's fast search and depth-first search's memory efficiency
- The iterative search algorithm is useful uninformed search when search space is large, and depth of goal node is unknown
- Advantages
  - It combines the benefits of BFS and DFS search algorithm in terms of fast search and memory efficiency
- Disadvantages
  - The main drawback of IDDFS is that it repeats all the work of the previous phase

Uninformed Search (Contd.)

- Example
  Following tree structure is showing the iterative deepening depth-first search. IDDFS algorithm performs various iterations until it does not find the goal node. The iteration performed by the algorithm is given as



1'st Iteration $\rightarrow$ A
2'nd Iteration $\rightarrow$ A, B, C
3'rd Iteration $\rightarrow$ A, B, D, E, C, F, G
4'th Iteration $\rightarrow$ A, B, D, H, I, E, C, F, K, G

```
function ITERATIVE-DEEPENING-SEARCH(problem) returns a solution sequence
    inputs: problem, a problem

    for depth ← 0 to ∞ do
        if DEPTH-LIMITED-SEARCH(problem, depth) succeeds then return its result
    end
    return failure
```

Thank You