SREENIVASA INSTITUTE OF TECHNOLOGY AND MANAGEMENT STUDIES, CHITTOOR

(AUTONOMOUS)

Approved by AICTE, New Delhi, Affiliated to JNTUA, Ananthapuramu.

LECTURE NOTES

Subject Name : Artificial Intelligence

Year / Branch : II Year/CSE(AI)

Regulation : R23

Prepared by : Dr. Munipraveena Rela

Sam Altman: **"AI won't replace humans, but humans who use AI will replace those who don't."**

1. Vision, Mission, POs, PEOs, PSOs

INSTITUTE VISION

To emerge as a Centre of Excellence for Learning and Research in the domains of engineering, computing and management.

INSTITUTE MISSION

- Provide congenial academic ambience with state-art of resources for learning and research.
- Ignite the students to acquire self-reliance in the latest technologies.
- Unleash and encourage the innate potential and creativity of students.
- Inculcate confidence to face and experience new challenges.
- Foster enterprising spirit among students.
- Work collaboratively with technical Institutes / Universities / Industries of National and International repute.

DEPARTMENT VISION

To contribute for the society through excellence in Computer Science and Engineering (Artificial Intelligence) with a deep passion for wisdom, culture and values.

DEPARTMENT MISSION

M1: Provide congenial academic ambience with necessary infrastructure and learning resources. M2: Inculcate confidence to face and experience new challenges from industry and society.

M3: Ignite the students to acquire self-reliance in State-of-the-Art Technologies.

M4: Foster Enterprising spirit among students.

PROGRAMME OUTCOMES (PO's)

On Successful completion, the graduate will be able to,

PO1. Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2. Problem analysis: Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4. Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5. Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9. Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11. Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one‟s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSO's)

On Successful completion, the graduate will be able to,

PSO1: Demonstrate and analyse the mathematical methodologies to develop computer programs using networking, web design, big data analytics, machine learning algorithms, data science, neural networks and deep learning techniques.

PSO2: Deploy the artificial intelligence techniques to perform human intelligence in creating innovative.

PROGRAMME EDUCATIONAL OBJECTIVES (PEOs)

After few years of graduation the, graduates of Computer Science and Engineering (Artificial Intelligence) shall

PEO1: Expertise with computer science and Engineering, artificial intelligence and machine learning disciplines through quality studies, enabling success in IT industries. (Professional Competency)

PEO2: Establish start-up companies or employed in reputed computing industries or government sectors or pursue higher studies in the domain of CSE (AI) (Successful Career Goals)

PEO3: Enhance knowledge by updating advanced technological concept for facing the rapidly changing world and contribute to society through innovations and creativity. (Continuing Education and Contribution to Society)

2. Syllabus

II B.Tech. - III Semester

23CAI231T	**ARTIFICIAL INTELLIGENCE**	**L T P C**
		3 0 0 3

PRE-REQUISITES:

- Knowledge in Computer Programming.
- A course on "Mathematical Foundations of Computer Science".
- Background in linear algebra, data structures and algorithms, and probability.

COURSE EDUCATIONAL OBJECTIVES:

1. To study the concepts of Artificial Intelligence.
2. To learn the methods of solving problems using Artificial Intelligence.
3. To introduce the concepts of Expert Systems.
4. To understand the applications of AI, namely game playing, theorem proving, and machine learning.
5. To learn different knowledge representation techniques

UNIT I: INTRODUCTION TO AI (9)

AI problems, foundation of AI and history of AI intelligent agents: Agents and environments, the concept of rationality, the nature of environments, structure of agents, problem solving agents, problem formulation.

UNIT 2: SEARCHING (9)

Searching for solutions, uniformed search strategies – Breadth first search, depth first Search. Search with partial information (Heuristic search) Hill climbing, A* ,AO* Algorithms, Problem reduction, Game Playing-Adversial search, Games, mini-max algorithm, optimal decisions in multiplayer games, Problem in Game playing, Alpha-Beta pruning, Evaluation functions.

UNIT 3: REPRESENTATION OF KNOWLEDGE (9)

Knowledge representation issues, predicate logic- logic programming, semantic nets- frames and inheritance, constraint propagation, representing knowledge using rules, rules based deduction systems. Reasoning under uncertainty, review of probability, Bayes' probabilistic interferences and dempstershafer theory.

UNIT 4: LOGIC CONCEPTS (9)

First order logic. Inference in first order logic, propositional vs. first order inference, unification & lifts forward chaining, Backward chaining, Resolution, Learning from observation Inductive learning, Decision trees, Explanation based learning, Statistical Learning methods, Reinforcement Learning.

UNIT 5: EXPERT SYSTEMS (9)

Architecture of expert systems, Roles of expert systems – Knowledge Acquisition Meta knowledge Heuristics. Typical expert systems – MYCIN, DART, XCON: Expert systems shells.

Total Hours: 45

COURSE OUTCOMES:

On successful completion of the course, students will be able to		POs
CO1	Study the concepts of Artificial Intelligence.	**PO1, PO2**
CO2	Learn the methods of solving problems using Artificial Intelligence	**PO1, PO2, PO3, PO4, PO5**
CO3	Introduce the concepts of Expert Systems.	**PO1, PO2, PO3, PO4, PO5**
CO4	Understand the applications of AI, namely game playing, theorem proving, and machine learning.	**PO1, PO2, PO3, PO4, PO5**
CO5	Learn different knowledge representation techniques	**PO1, PO2, PO3, PO4**

TEXT BOOKS:

1. S. Russel and P. Norvig, "Artificial Intelligence – A Modern Approach", Second Edition, Pearson Education, 2007.
2. Kevin Night and Elaine Rich, Nair B., "Artificial Intelligence (SIE)", Mc Graw Hill, 2008.

REFERENCE BOOKS:

1. David Poole, Alan Mackworth, Randy Goebel, "Computational Intelligence: a logical approach", Oxford University Press, 1998.
2. G. Luger, "Artificial Intelligence: Structures and Strategies for complex problem solving", Sixth Edition, Pearson Education, 2009.
3. J. Nilsson, "Artificial Intelligence: A new Synthesis", 1st Edition, Elsevier Publishers, 1997.
4. Saroj Kaushik, "Artificial Intelligence", 2nd Edition, CENGAGE Learning, 2022.

REFERENCE WEBSITE:

1. https://ai.google/
2. https://swayam.gov.in/nd1_noc19_me71/preview

CO-PO MAPPING:

CO\ PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	-	-	-	-	-	-	-	-	-	-
CO2	3	3	2	2	2	-	-	-	-	-	-	-
CO3	3	2	2	2	2	-	-	-	-	-	-	-
CO4	3	2	2	2	2	-	-	-	-	-	-	-
CO5	3	2	2	2	-	-	-	-	-	-	-	-
CO*	**3**	**2**	**2**	**2**	**2**	**-**	**-**	**-**	**-**	**-**	**-**	**-**

Unit -1
Introduction

"The science of artificial intelligence is the science of making machines do things that would require intelligence if done by humans."— **Marvin Minsky**

Unit Overview

Artificial Intelligence (AI) is one of the most transformative technologies of the 21st century, enabling computers and machines to perform tasks that traditionally require human intelligence. This unit introduces the fundamental concepts of AI, including its definition, history, foundations, and major application areas. Students will explore the concept of intelligent agents, their interaction with environments, the principle of rationality, and different types of environments. The unit also covers the structure of intelligent agents, problem-solving agents, and the process of formulating problems for AI systems. These concepts provide the theoretical foundation required for understanding advanced AI techniques discussed in later units.

Objectives of the Unit

After studying this unit, students will be able to:

1. Understand the concept, scope, and significance of Artificial Intelligence.
2. Describe the historical evolution and foundations of AI.
3. Explain the characteristics and components of intelligent agents.
4. Differentiate between various types of environments in which agents operate.
5. Understand the concept of rationality in intelligent agents.

Learning Outcomes

Upon successful completion of this unit, students will be able to:

LO1: Define Artificial Intelligence and identify its major application domains.
LO2: Explain the historical milestones and foundational disciplines of AI.
LO3: Describe intelligent agents, their components, and interactions with environments.
LO4: Classify environments based on observability, determinism, dynamics, discreteness, and agent participation.
LO5: Explain the concept of rational agents and rational decision-making.

Importance of Studying this Unit

Studying this unit is important because it lays the conceptual foundation for the entire field of Artificial Intelligence. It helps students understand how intelligent systems perceive their surroundings, reason about available information, and make decisions to achieve specific goals. Knowledge of intelligent agents is fundamental to modern technologies such as autonomous vehicles, robotics, virtual assistants, recommendation systems, healthcare diagnostics, financial analytics, cyber security, and smart manufacturing. A clear understanding of AI fundamentals also prepares students to study advanced topics such as machine learning, deep learning, natural language processing, computer vision, and reinforcement learning.

1.1 Artificial Intelligence:

- **Artificial Intelligence** is the branch of computer science by which we can create intelligent machines which can behave like a human, think like humans, and able to make decisions.“
- Major AI textbooks define artificial intelligence as "the study and design of intelligent agents," where an **intelligent agent** is a system that **perceives** its **environment** and **takes actions** which maximize its chances of success.
- **John McCarthy**, who coined the term in 1956, defines it as "the science and engineering of making intelligent machines, especially intelligent computer programs."

The term AI is defined by each author in own ways which falls into 4 categories

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

Thinking Humanly	**Thinking Rationally**
“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)	“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)
Acting Humanly	**Acting Rationally**
“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)	“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)

Figure 1.1 Some definitions of artificial intelligence, organized into four categories.

- we see eight definitions of AI, laid out along two dimensions. The definitions on top are concerned with *thought processes* and *reasoning*, whereas the ones on the bottom address *behavior*.
- The definitions on the left measure success in terms of fidelity to *human* performance, whereas the ones on the right measure against an *ideal* performance measure, called **rationality**.
- A system is rational if it does the “right thing,” given what it knows.

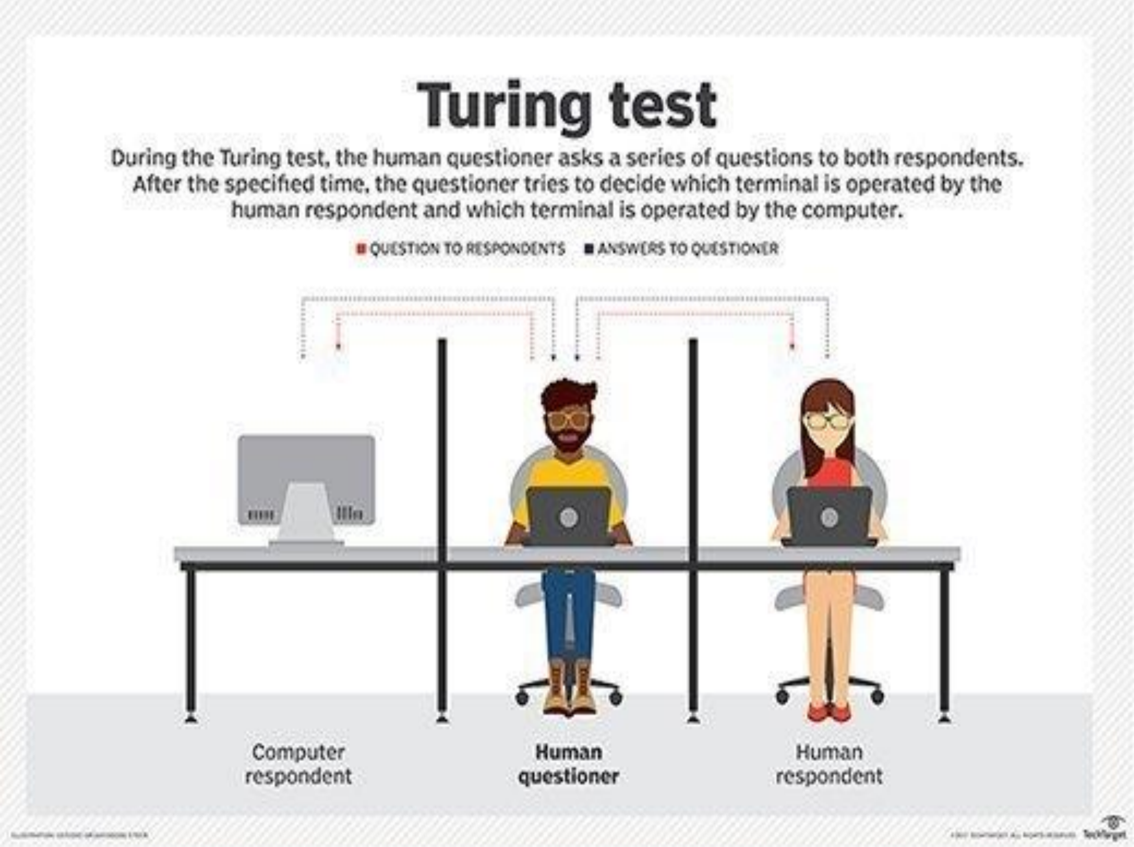
Acting Humanly: The Turing Test Approach

- Turing Test proposed by Alan Turing in 1950
- The computer is asked questions by a human interrogator.

- The computer passes the test if a human interrogator, after giving some written questions, cannot tell whether the responses come from a person or computer. Programming a computer to pass, the computer need to have the following capabilities:

Natural language processing to enable it to communicate successfully in English.
Knowledge representation to store what it knows or hears
Automated reasoning to use the stored information to answer questions and to draw new conclusions.
Machine learning to adapt to new circumstances and to detect and extrapolate patterns.

- **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 as input. To pass the total Turing Test, the computer will need
- **Computer vision** to perceive objects, and
- **Robotics** to manipulate objects and move about.



Thinking humanly: The cognitive modeling approach

- To say that a given program thinks like a human, we must have some way of determining how humans think
- We need to get inside actual working of the human mind, this can be done:
- Through introspection – trying to capture our own 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 behavior matches corresponding human behavior, that is evidence that some of the program's mechanisms could also be operating in humans.

- Allen Newell and Herbert Simon, who developed GPS, the ―General Problem Solver‖ tried to trace the reasoning steps to traces of human subjects solving the same problems. 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

Thinking rationally: The "laws of thought approach"

- The Greek philosopher Aristotle was one of the first to attempt to codify ―right thinking that is irrefutable (ie. Impossible to deny) reasoning processes. His syllogism provided patterns for argument structures that always yielded correct conclusions when given correct premises
- for example, Socrates is a man; all men are mortal; therefore Socrates is mortal.
- These laws of thought were supposed to govern the operation of the mind; their study initiated a field called logic.
- The **logicist** tradition within artificial intelligence is used to build programs to create intelligent systems.

Acting rationally: The rational agent approach

- An **agent** is just something that acts (*agent* comes from the Latin *agere*, to do). Of course, all computer programs do something, but
- Computer agents are not mere programs, but they are expected to have the following attributes also:
  (a) Operating under autonomous control,
  (b) Perceiving their environment,
  (c) Persisting over a prolonged time period,
  (d) Adapting to change and create and pursue goals.

1.2 Foundations of AI

- A **rational agent** is one that acts so as to achieve the best outcome or, when there is uncertainty, the best expected outcome.
- The disciplines that contributed ideas, viewpoints, and techniques to AI while addressing some questions.
- **Philosophy**
- Can formal rules be used to draw valid conclusions?
- How does the mental mind arise from a physical brain?
- Where does knowledge come from?
- How does knowledge lead to action?
- **Mathematics**
- What are the formal rules to draw valid conclusions?
- What can be computed?
- How do we reason with uncertain information?
- **Economics**
- How should we make decisions so as to maximize payoff?
- How should we do this when others may not go along?
- How should we do this when the payoff may be far in the future?
- **Neuroscience**

- How do brains process information?
- **Psychology**
- How do humans and animals think and act?
- **Computer engineering**
- How can we build an efficient computer?
- **Control theory and cybernetics**
- How can artifacts operate under their own control?

1.3 Levels of AI (types of artificial intelligence)

- ANI: Artificial narrow intelligence
- AGI: Artificial general intelligence
- ASI: Artificial super intelligence

Narrow AI (Weak AI)

- This is the most common type of AI currently in use.
- It's designed to perform a specific task extremely well, often better than humans, but it lacks general intelligence.
- Focuses on one task and cannot perform beyond its limitations (common in our daily lives)
- makes use of NLP to perform tasks
- Examples include:
- Email Spam filters
- Recommendation systems
- Facial recognition software
- Robots that perform repetitive tasks on assembly lines
- Virtual assistants like Siri or Alexa
- IBMS Watson, Self driving cars

General AI (Strong AI):

This level of AI would possess human-level cognitive abilities, meaning it could learn, understand, and apply knowledge across various tasks, just like a human. General AI could potentially learn new things on its own and apply its knowledge to new situations. This level of AI is currently theoretical and not yet achieved. strong AI or Deep AI Can understand and learn any intellectual task that a human being can (researchers are striving to reach strong AI)machines can mimic human intelligence ability to apply their intelligence to solve problems. Properties of Deep AI are: recognition, recall, imagination Ex: Advanced robotics

ASI Artificial super intelligence

Surpasses human intelligence and can perform any task better than a human (still a concept)level where machines super passes human capabilities and become self-aware, able to build emotions of its own, better than human in sports, math, science etc., in movies only

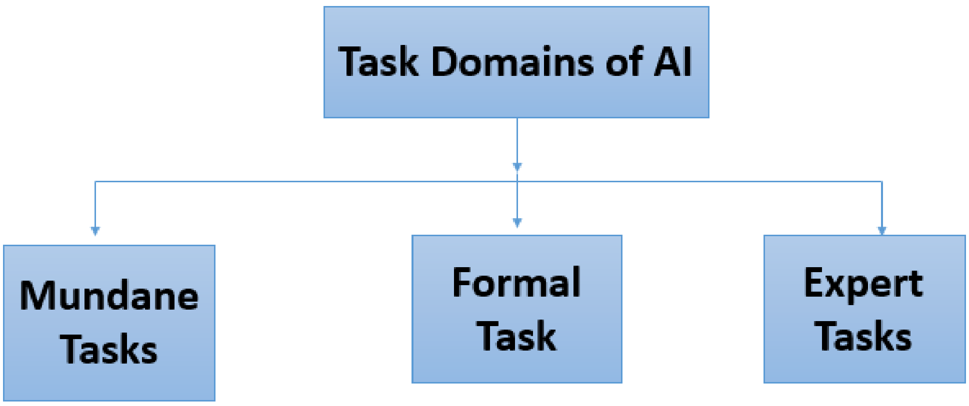
1.4 History of AI

- Year 1943: The first work which is now recognized as AI was done by Warren McCulloch and Walter pits in 1943. They proposed a model of artificial neurons.
- Year 1949: Donald Hebb demonstrated an updating rule for modifying the connection strength between neurons. His rule is now called Hebbian learning.
- Year 1950: Alan Turing publishes "Computing Machinery and Intelligence" in which he proposed a test called a Turing test.The birth of artificial intelligence (1956)
- Year 1955: the "first artificial intelligence program was created , Which was named as "Logic

Theorist". This program had proved 38 of 52 Mathematics theorems, and find new and more elegant proofs for some theorems.

- Year 1956: The word "Artificial Intelligence" first adopted by American Computer scientist John McCarthy at the Dartmouth Conference. For the first time, AI coined as an academic field.At that time high-level computer languages such as FORTRAN, LISP, or COBOL were invented.
- Year 1966: The researchers emphasized developing algorithms which can solve mathematical problems. the first chatbot was created in 1966, which was named as ELIZA.
- Year 1972: The first intelligent humanoid robot was built in Japan which was named as WABOT-1.
- The first AI winter (1974-1980) The duration between years 1974 to 1980 was the first AI winter duration. AI winter refers to the time period where computer scientist dealt with a severe shortage of funding from government for AI researches.
- Year 1980: After AI winter duration, AI came back with "Expert System". Expert systems were programmed that emulate the decision-making ability of a human expert.In the Year 1980, the first national conference of the American Association of Artificial Intelligence was held at Stanford University.
- The second AI winter (1987-1993)The duration between the years 1987 to 1993 was the second AI Winter duration.Again Investors and government stopped in funding for AI research as due to high cost but not efficient result. The expert system such as XCON was very cost effective
- Year 1997: In the year 1997, IBM Deep Blue beats world chess champion, Gary Kasparov, and became the first computer to beat a world chess champion.
- Year 2002: for the first time, AI entered the home in the form of Roomba, a vacuum cleaner.Year 2006: AI came in the Business world till the year 2006. Companies like Facebook, Twitter, and Netflix also started using AI.
- Deep learning, big data and artificial general intelligence (2011-present) Year 2011: In the year 2011, IBM's Watson won jeopardy, a quiz show, where it had to solve the complex questions as well as riddles. Watson had proved that it could understand natural language and can solve tricky questions quickly.Year 2012: Google has launched an Android app feature "Google now", which was able to provide information to the user as a prediction.Year 2014: In the year 2014, Chatbot "Eugene Goostman" won a competition in the famous "Turing test."Year 2018: The "Project Debater" from IBM debated on complex topics with two master debaters and also performed extremely well.

1.5 AI Problems or Task Domains of AI



Mundane tasks	Formal tasks	Expert tasks
Perception – Computer Vision – Speech, Voice	Games – Go – Chess (Deep Blue) – Ckeckers	Engineering – Design – Fault Finding – Manufacturing – Monitoring
Natural Language Processing – Understanding – Language Generation – Language Translation	Mathematics – Geometry – Logic – Integration and Differentiation	Scientific Analysis
Common Sense Reasoning	Theorem Proving	Financial Analysis
Planning		Medical Diagnosis
Robot Control		

1.6 The State of the Art (the application of Artificial Intelligence)

- **Robotic vehicles:** A driverless robotic car named STANLEY raced through the rough terrain of the Mojave dessert at 22 mph, finishing the 132-mile distance first to win the 2005 DARPA Grand Challenge. STANLEY is a Volkswagen Touareg outfitted with cameras, radar, and laser range finders to sense the environment and onboard software to control the steering, braking, and acceleration.
- **Speech recognition:** A traveler calling United Airlines to book a flight can have the entire conversation guided by an automated speech recognition and dialog management system.Autonomous planning and scheduling: A hundred million miles from Earth, NASA's Remote Agent program became the first on-board autonomous planning program to control the scheduling of operations for a spacecraft (Jonsson et al., 2000). REMOTE AGENT generated plans from high-level goals specified from the ground and monitored the execution of those plans—detecting, diagnosing, and recovering from problems as they occurred.
- **Game playing:** IBM's DEEP BLUE became the first computer program to defeat the world champion(Garry Kasparov) in a chess match. Human champions studied Kasparov's loss and were able to draw a few matches in subsequent years, but the most recent human-computer matches have been won convincingly by the computer.
- **Spam filtering:** Each day, learning algorithms classify over a billion messages as spam, saving the recipient from having to waste time deleting what, for many users, could comprise 80% or 90% of all messages, if not classified away by algorithms. Because the spammers are continually updating their tactics, it is difficult for a static programmed approach to keep up, and learning algorithms work best
- **Logistics planning:** During the Persian Gulf crisis of 1991, U.S. forces deployed a Dynamic Analysis and Replanning Tool, DART (Cross and Walker, 1994), to do automated logistics planning and scheduling for transportation. This involved up to 50,000 vehicles, cargo, and people at a time, and had to account for starting points, destinations, routes, and conflict resolution among all parameters. The AI planning techniques generated in hours a plan that would have taken weeks with older methods. The Defense Advanced Research Project Agency (DARPA) stated that this

single application more than paid back DARPA's 30-year investment in AI.

- **Robotics:** The iRobot Corporation has sold over two million Roomba robotic vacuum cleaners for home use. The company also deploys the more rugged PackBot to Iraq and Afghanistan, where it is used to handle hazardous materials, clear explosives, and identify the location of snipers.
- **Machine Translation**: A computer program automatically translates from Arabic to English. The program uses a statistical model built from examples of Arabic-to-English translations and from examples of English text totaling two trillion words (Brants et al., 2007). None of the computer scientists on the team speak Arabic, but they do understand statistics and machine learning algorithms.

1.7 Advantages of Artificial Intelligence

1. **High Accuracy with less errors:** Artificial Intelligence can significantly reduce errors and increase accuracy and precision. The decisions taken by AI in every step is decided by information previously gathered and a certain set of algorithms. When programmed properly, these errors can be reduced to null.
2. **High-Speed:** AI systems can be of very high-speed and fast-decision making, because of that AI systems can beat a chess champion in the Chess game.
3. **High reliability:** AI machines are highly reliable and can perform the same action multiple times with high accuracy.
4. **Useful for risky areas:** Another big advantage of AI is that humans can overcome many risks by letting AI robots do them for us. Whether it be defusing a bomb, going to space, exploring the deepest parts of oceans, machines with metal bodies are resistant in nature and can survive unfriendly atmospheres. Moreover, they can provide accurate work with greater responsibility and not wear out easily.
5. **24x7 Availability**: There are many studies that show humans are productive only about 3 to 4 hours in a day. Humans also need breaks and time offs to balance their work life and personal life. But AI can work endlessly without breaks. They think much faster than humans and perform multiple tasks at a time with accurate results. They can even handle tedious repetitive jobs easily with the help of AI algorithms.
6. **Digital Assistance:** Some of the most technologically advanced companies engage with users using digital assistants, which eliminates the need for human personnel. Many websites utilize digital assistants to deliver user-requested content. We can discuss our search with them in conversation. Some chatbots are built in a way that makes it difficult to tell whether we are conversing with a human or a chatbot. We all know that businesses have a customer service crew that must address the doubts and concerns of the patrons. Businesses can create a chatbot or voice bot that can answer all of their clients' questions using AI.AI technology is currently used by various E-commerce websites to show the products as per customer requirement.
7. **New Inventions:** In practically every field, AI is the driving force behind numerous innovations that will support humans in resolving the majority of challenging issues.For instance, recent advances in AI-based technologies have allowed doctors to detect cancer at an earlier stage.
8. **Unbiased Decisions:** Human beings are driven by emotions, whether we like it or not. AI on the other hand, is devoid of emotions and highly practical and rational in its approach. A huge

advantage of Artificial Intelligence is that it doesn't have any biased views, which ensures more accurate decision-making.

9. **Perform Repetitive Jobs:** We will be doing a lot of repetitive tasks as part of our daily work, such as checking documents for flaws and mailing thank-you notes, among other things. We may use artificial intelligence to efficiently automate these menial chores and even eliminate "boring" tasks for people, allowing them to focus on being more creative. Example: In banks, it's common to see multiple document checks to obtain a loan, which is a time-consuming task for the bank's owner. The owner can speed up the document verification process for the advantage of both the clients and the owner by using AI Cognitive Automation.

10. **Daily Applications:** Today, our everyday lives are entirely dependent on mobile devices and the internet. We utilize a variety of apps, including Google Maps, Alexa, Siri, Cortana on Windows, OK Google, taking selfies, making calls, responding to emails, etc. With the use of various AI-based techniques, we can also anticipate today's weather and the days ahead.Example: About 20 years ago, you must have asked someone who had already been there for instructions when you were planning a trip. All you need to do now is ask Google where Bangalore is. The best route between you and Bangalore will be displayed, along with Bangalore's location, on a Google map.

1.8 Disadvantages of Artificial Intelligence

1. **High Costs:** The ability to create a machine that can simulate human intelligence is no small feat. It requires plenty of time and resources and can cost a huge deal of money. AI also needs to operate on the latest hardware and software to stay updated and meet the latest requirements, thus making it quite costly.

2. **No creativity:** A big disadvantage of AI is that it cannot learn to think outside the box. AI is capable of learning over time with pre-fed data and past experiences, but cannot be creative in its approach. A classic example is the bot Quill who can write Forbes earning reports. These reports only contain data and facts already provided to the bot. Although it is impressive that a bot can write an article on its own, it lacks the human touch present in other Forbes articles.

3. **Unemployment:** One application of artificial intelligence is a robot, which is displacing occupations and increasing unemployment (in a few cases). Therefore, some claim that there is always a chance of unemployment as a result of chatbots and robots replacing humans. For instance, robots are frequently utilized to replace human resources in manufacturing businesses in some more technologically advanced nations like Japan. This is not always the case, though, as it creates additional opportunities for humans to work while also replacing humans in order to increase efficiency.

4. **Make Humans Lazy:** AI applications automate the majority of tedious and repetitive tasks. Since we do not have to memorize things or solve puzzles to get the job done, we tend to use our brains less and less. This addiction to AI can cause problems to future generations.With the increment of technology, people are getting more dependent on devices and hence they are losing their mental capabilities.

5. **No Ethics:** Ethics and morality are important human features that can be difficult to incorporate into an AI. The rapid progress of AI has raised a number of concerns that one day, AI will grow uncontrollably, and eventually wipe out humanity. This moment is referred to as the AI singularity.

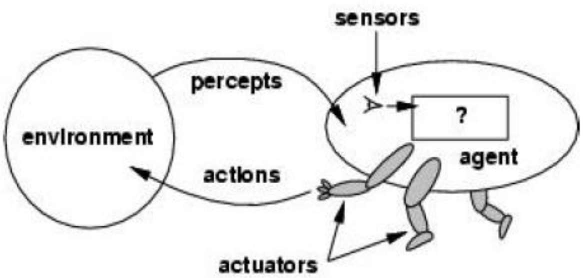
6. **Emotionless:** Since early childhood, we have been taught that neither computers nor other

machines have feelings. Humans function as a team, and team management is essential for achieving goals. However, there is no denying that robots are superior to humans when functioning effectively, but it is also true that human connections, which form the basis of teams, cannot be replaced by computers.

7. **No Improvement:** Artificial intelligence is a technology based on pre-loaded facts and experience. AI is proficient at repeatedly carrying out the same task, but if we want any adjustments or improvements, we must manually alter the codes. AI cannot be accessed and utilized similar to human intelligence, but it can store infinite data. Machines can only complete tasks they have been developed or programmed for; if they are asked to complete anything else, they frequently fail or provide useless results, which can have significant negative effects. Thus, we are unable to make anything conventional.

1.9 AGENTS

An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators. This simple idea is illustrated in Figure.



A **human agent** has eyes, ears, and other organs for sensors and hands, legs, mouth, and other body parts for actuators.

A **robotic agent** might have cameras and infrared range finders for sensors and various motors for actuators.

A **software agent** receives keystrokes, file contents, and network packets as sensory inputs and acts on the environment by displaying on the screen, writing files, and sending network packets.

Percept: Percept is used to refer to the agent's perceptual inputs at any given instant.

Percept Sequence: An agent's percept sequence is the complete history of everything the agent has ever perceived. In general an agent's choice of action at any given instant can depend on the entire percept sequence observed to date.

Agent function: Mathematically speaking, an agent's behavior is described by the agent function that maps any given percept sequence to an action.

$$f : P^* \rightarrow A$$

for most agents, Agent function would be a very large tableThe agent program runs on the physical architecture to produce agent function(f).

agent = architecture + program

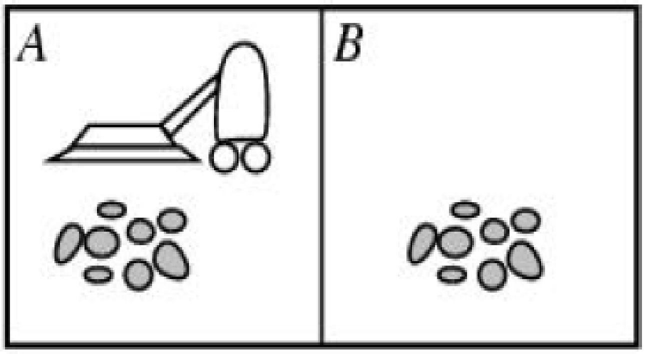
Agent program: The agent function for an artificial agent will be implemented by an agent program. The agent function is an abstract mathematical description; The agent program is a concrete implementation, running on the agent architecture.

Example-the vacuum-cleaner world. This particular world has just two locations: squares A and B. The vacuum agent perceives which square it is in and whether there is dirt in the square. It can choose to move left, move right, suck up the dirt, or do nothing.

One very simple agent function is the following:

if the current square is dirty, then suck,

otherwise, it move to the other square.



Percepts: location and status, [A , Dirty]
Actions: Left, Right, Suck, NoOp

Partial tabulation of a simple agent function for the vacuum cleaner world

Percept Sequence	Action
[A, Clean]	Right
[A, Dirty]	Suck
[B, Clean]	Left
[B, Dirty]	Suck
[A, Clean], [A, Clean]	Right
[A, Clean], [A, Dirty]	Suck
.....	.....

Example of Agent program

function Reflex-VACUUM-AGENT ([locations, status]) returns an action
if status = Dirty then return Suck
else if location = A then return Right
elseif location = B then return Left

1.10 Good Behavior: The concept of Rationality

A rational agent is one that does the right thing-based on what it can perceive and the actions it can perform. every entry in the table for the agent function is filled out correctly. The right action is the one that will cause the agent to be most successful.
A performance measure represents the criterion for success of an agent's behavior. When an agent is placed in an environment, it generates a sequence of actions according to the percepts it receives.

This sequence of actions causes the environment to go through a sequence of states. If the sequence is desirable, then the agent has performed well.

E.g., performance measure of a vacuum-cleaner agent could be amount of dirt cleaned up, amount of time taken, amount of electricity consumed, amount of noise generated, etc.

Rationality: The rationality at any given time depends on four things:

1. The performance measure that defines the criterion of success.
2. The agent's prior knowledge of the environment.
3. The actions that the agent can perform.
4. The agent's percept sequence to date.

Definition of a rational agent: For each possible percept sequence, a rational agent should select an action that is expected to maximize its performance measure, given the evidence provided by the percept sequence and some built-in knowledge the agent has about the environment.

• **Is a vacuum-agent rational? (Depends!)**

• Performance measure: awards one point for each clean square at each time step, over a lifetime of 1000 time steps. Geography of the environment is known a priori (two squares, A is on left and B is on right, etc.). Clean squares stay clean and sucking cleans the current square. The Left and Right actions move the agent left and right. The only available actions are Left, Right, and Suck. The agent correctly perceives its location and whether it contains dirt. We claim that under these circumstances the agent is indeed rational

The same agent would be irrational under different circumstances.

For ex., once all the dirt is cleaned up, the agent will oscillate needlessly back and forth; if the performance measure includes a penalty of one point for each movement left or right, the agent will fare poorly. A better agent for this case would do nothing once it is sure that all the squares are clean. If clean squares can become dirty again, the agent should occasionally check and re-clean them if needed. If the geography of the environment is unknown, the agent will need to explore it rather than stick to squares A and B.

1.11 Task environments

Task environments, which are essentially the "problems" to which rational agents are the "solutions."

Specifying the task environment:

The rationality of the simple agent, needs specification of

- the performance measure
- the environment
- the agent's actuators
- Sensors.

PEAS

All these are grouped together under the heading of the task environment. We call this the PEAS (Performance, Environment, Actuators, Sensors) description. In designing an agent, the first step must always be to specify the task environment as fully as possible.

PEAS Example: Autonomous taxi

- Performance measure
  - Safe, fast, legal, comfortable trip, maximize profits
- Environment
  - Roads, other traffic, pedestrians, customers
- Actuators
  - Steering wheel, accelerator, brake, signal, horn
- Sensors
  - Cameras, LIDAR, speedometer, GPS, odometer, engine sensors, keyboard

PEAS

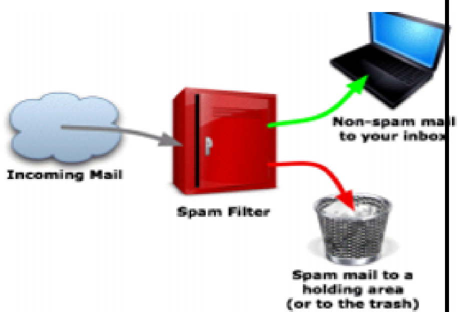
How about a vacuum cleaner?

iRobot Roomba series

- Performance: cleanness, efficiency: distance traveled to clean, battery life, security.
- Environment: room, table, wood floor, carpet, different obstacles.
- Actuators: wheels, different brushes, vacuum extractor.
- Sensors: camera, dirt detection sensor, cliff sensor, bump sensors, infrared wall sensors.

Another PEAS example: Spam filter

- Performance measure
  - Minimizing false positives, false negatives
- Environment
  - A user's email account, email server
- Actuators
  - Mark as spam, delete, etc.
- Sensors
  - Incoming messages, other information about user's account



PEAS for a medical diagnosis system

- Performance measure: Healthy patient, minimize costs, lawsuits
- Environment: Patient, hospital, staff
- Actuators: Screen display (questions, tests, diagnoses, treatments, referrals)
- Sensors: Keyboard (entry of symptoms, findings, patient's answers)

PEAS for a part-picking robot

- Performance measure: Percentage of parts in correct bins
- Environment: Conveyor belt with parts, bins
- Actuators: Jointed arm and hand
- Sensors: Camera, joint angle sensors

1.12 Properties of task environments (types of task environment) (Environment types)

1. Fully observable vs. partially observable
2. Deterministic vs. stochastic
3. Episodic vs. sequential
4. Static vs. dynamic
5. Discrete vs. continuous
6. Single agent vs. multi agent

Fully observable vs. partially observable: If an agent's sensors give it access to the complete state of the environment at each point in time, then we say that the task environment is fully observable. A task environment is effectively fully observable if the sensors detect all aspects that are relevant to the choice of action;

An environment might be partially observable because of noisy and inaccurate sensors or because parts of the state are simply missing from the sensor data. Fully observable environments are convenient, because the agent need not maintain any internal state to keep track of the world– Playing chess? Fully.– Taxi driving? Partially.

If the agent has no sensors at all then the environment is unobservable For example , a vacuum agent with only a local dirt sensor cannot tell whether there is dirt in other squares, and an automated taxi cannot see what other drivers are thinking.

Deterministic vs. stochastic: If the next state of the environment is completely determined by the current state and the action executed by the agent, then we say the environment is deterministic; Otherwise, it is stochastic. If the environment is deterministic except for the actions of other agents, then the environment is strategic. In principle, an agent need not worry about uncertainty in a fully observable, deterministic environment Taxi driving is stochastic, because one can never predict the behavior of traffic exactly, and vacuum world is deterministic.

Episodic vs. sequential: In an episodic task environment, the agent's experience is divided into atomic episodes. Each episode consists of the agent perceiving and then performing a single action. The next episode does not depend on the actions taken in previous episodes. For example, an agent that has to spot defective parts on an assembly line bases each decision on the current part, regardless of previous decisions; moreover, the current decision doesn't affect whether the next part is defective. In sequential environments, on the other hand, the current decision Could affect all future decisions.Chess and taxi driving are sequential : in both cases, short-term actions can have long term consequences. Episodic environments are much simpler than sequential environments because the agent does not need to think ahead.

Discrete vs. continuous: The discrete/continuous distinction can be applied to the state of the environment, to the way time is handled, and to the percepts and actions of the agent. For example, a discrete-state environment such as a chess game has a finite number of distinct states. Chess also has a discrete set of percepts and actions. Taxi driving is a continuous- state and continuous-time problem: The speed and location of the taxi and of the other vehicles sweep through a range of continuous values and do so smoothly over time. Taxi-driving actions are also continuous (steering angles, etc.)

Single agent vs. multiagent:if a single agent is present in the environment then it is Single agent environment, if more than one agent present then it is multiagent. An agent solving a crossword puzzle by itself is clearly in a single-agent environment, Where as an agent playing chess is in a two-agent environment.Multiagent is further classified in to two ways Competitive multiagent environment, Cooperative multiagent environment. Chess is a Competitive multiagent environment, Taxi driving environment is a Cooperative multiagent environment

Static Vs dynamic : If the environment can change while an agent is thinking, then we say the environment is dynamic for that agent; otherwise it is static. Static environments are easy to deal with because the agent need not keep looking at the world while it is deciding on an action. Dynamic environments, on the other hand, are continuously asking the agent what it wants to do. If the environment itself does not change with the passage of time but the agent's performance score does, then we say the environment is semidynamic. Taxi driving : dynamic: the other cars and taxi itself keep moving while the driving algorithm dithers about what to do next Chess when played with a clock is semi dynamic Crossword puzzles are static

1.13 THE STRUCTURE OF AGENTS(Agent Programs)

The job of AI is to design an agent program that implements the agent function that maps percepts to actions. this program will run on some sort of computing device with physical sensors and actuators—we call this the architecture: agent = architecture + program

The program we choose has to be appropriate for the architecture.

Ex. If the program is going to recommend actions like Walk, the architecture should better have legs. The architecture might be just an ordinary PC, or it might be a robotic car with several onboard computers, cameras, and other sensors. 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 actuators as they are generated.

Agent program that we design, take the current percept as input from the sensors and return an action to the actuators. agent program takes the current percept as input where as the agent function takes the entire percept history.A trivial agent program that keeps track of the percept sequence and then uses it to index into a table of actions to decide what to do.(Table driven agent) is as follows

```
function TABLE-DRIVEN-AGENT(percept ) returns an action
persistent: percepts, a sequence, initially empty
  table, a table of actions, indexed by percept sequences, initially fully
  specified
append percept to the end of percepts
action ←LOOKUP(percepts, table)
return action
```

To build a rational agent in this way, we as designers must construct a table that contains the appropriate action for every possible percept sequence. Table driven approach to agent construction leads to failure. Let P be the set of possible percepts and let T be the lifetime of the agent then The lookup table will contain $\sum_{t=1}^{T} |\mathcal{P}|^t$ entries

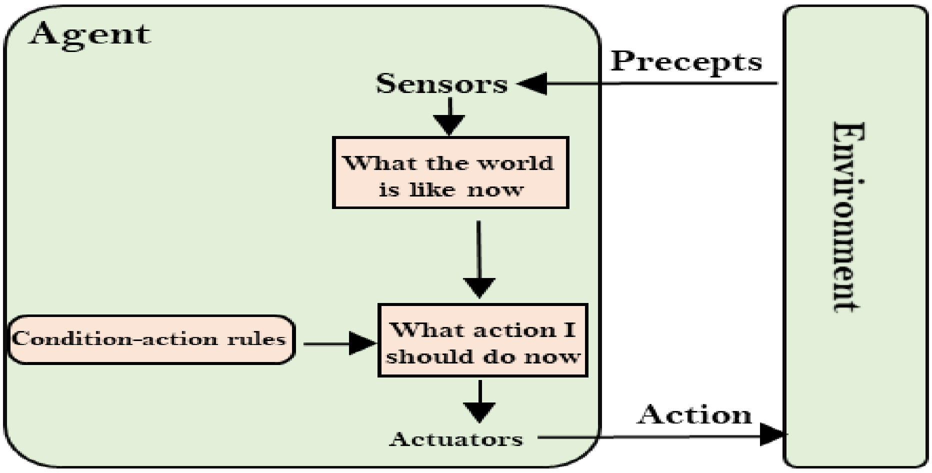
Consider the automated taxi: the visual input from a single camera comes in at the rate of roughly 27 megabytes per second (30 frames per second, 640×480 pixels with 24 bits of color information) This gives a lookup table for automated taxi with over 10^250,000,000,000 entries for one hour of driving. lookup table for chess would have at least 10^150 entries.

1.14 Types of agent programs

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

Simple Reflex Agent: The simplest kind of agent is the simple reflex agent. These agents select actions on the basis of the current percept, ignoring the rest of the percept history. These agents only succeed in the fully observable environment. For example, the vacuum agent whose agent function is tabulated, a simple reflex agent, because its decision is based only on the current location and on whether that contains dirt. Select action on the basis of only the current percept. E.g. the vacuum-agent

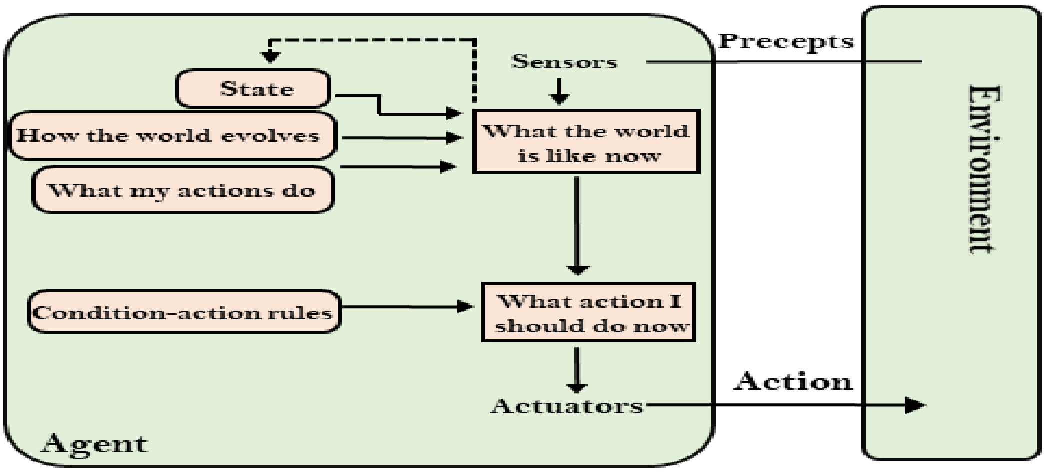
Simple reflex agent is implemented through condition-action rules which means it maps the current state to action Example: If dirty then suck Example: if car-in-front-is-breaking then initiate-breaking



function SIMPLE-REFLEX-AGENT(*percept*) **returns** an action
persistent: *rules*, a set of condition–action rules

state ← INTERPRET-INPUT(*percept*)
rule ← RULE-MATCH(*state*, *rules*)
action ← *rule*.ACTION
return *action*

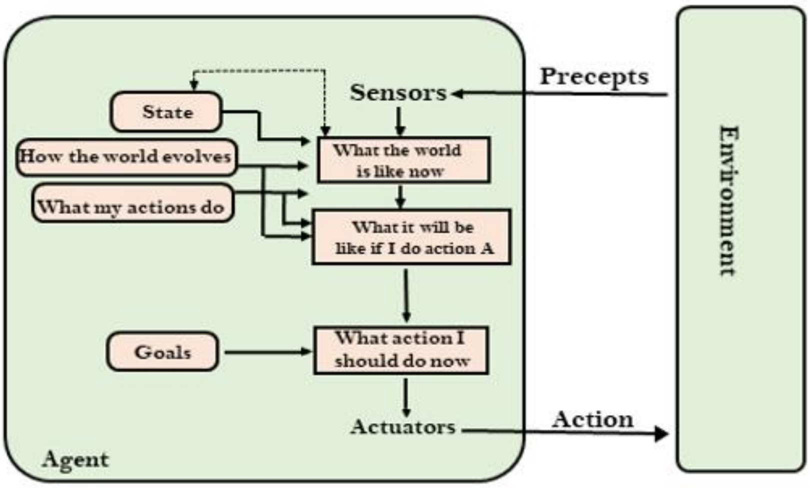
Model-based reflex agent: The Model-based agent can work in a partially observable environment, and track the situation. A model-based agent has two important factors: Model: It is knowledge about "how things happen in the world," so it is called a Model-based agent. Internal State: It is a representation 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. Updating the agent state requires information about: How the world evolves because of other agents. How the agent's action affects the world.



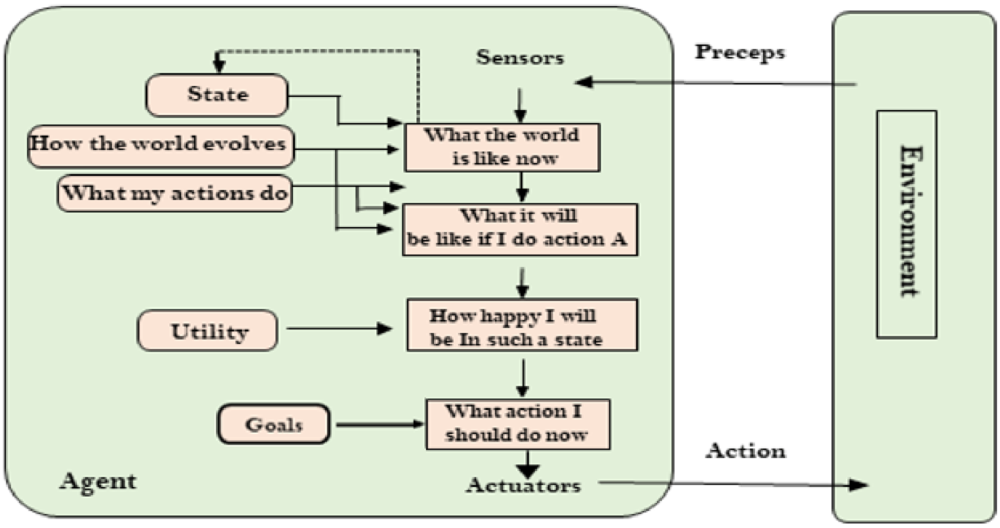
function MODEL-BASED-REFLEX-AGENT(*percept*) **returns** an action
  persistent: *state*, the agent's current conception of the world state
    model, a description of how the next state depends on current state and action
    rules, a set of condition–action rules
    action, the most recent action,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

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

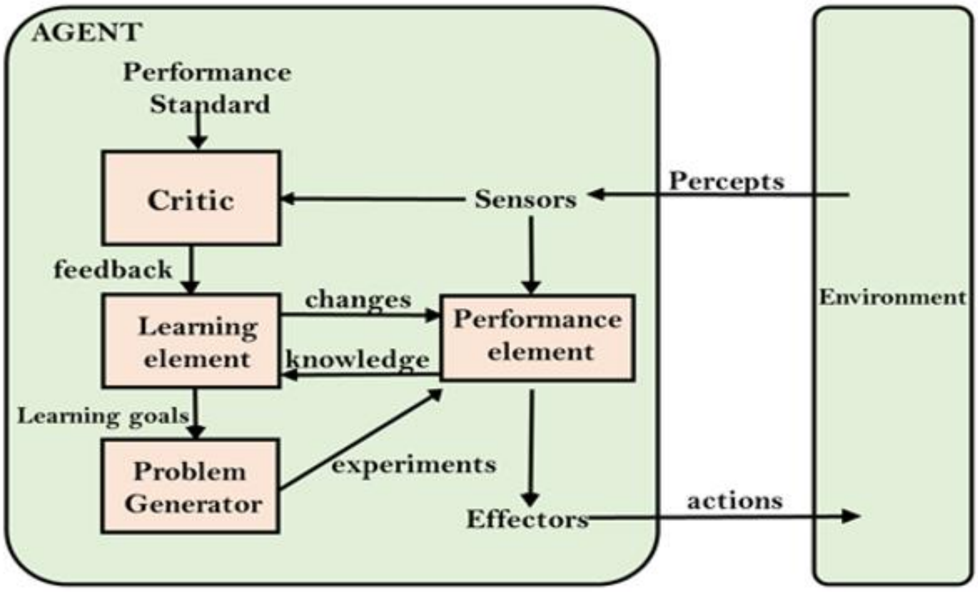
Goal-based agents: The knowledge of the current state of environment is not always sufficient to decide for an agent to what to do. The agent needs to know its goal which describes desirable situations. Goal-based agents expand the capabilities of the model-based agent by having the "goal" information. 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



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. The utility function maps each state to a real number to check how efficiently each action achieves the goals.



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



1.15 Problem-solving agents (PSA)

It's a goal-based agent. It decides what to do by finding sequences of actions that lead to desirable states. Intelligent agents are supposed to adopt a goal and aim to satisfy so as to maximize the performance measure. Goal formulation is the first step in problem solving. Goal formulation based on the current situation and the agent's performance measure. The agent has to find out which sequence of actions will get to a goal state. Agent needs to decide what sorts of actions and states are to be considered.
Problem formulation is the process of deciding what actions and states to consider, given a goal. Traveling agent - Unless the agent is very familiar and has the previous knowledge of the geography of destination, it cannot make proper decision to reach destination. Therefore, we need to search for the possible sequences? The process of examining different possible sequences of actions that reaches the goal is called Search. A search algorithm takes a problem as input and returns a solution in the form of an action sequence. Once a solution is found, then the recommended actions can be carried out, this phase is known as execution
The complete process of an agent to solve the problems can be shown as "formulate, search, execute" design for the agent. A simple problem solving agent. Formulate a goal and a problem. Search for a sequence of actions to solve the problem. Execute the actions one at a time. When

complete, formulate another goal and continue. Note that while executing the sequence, the agent ignores its percepts assuming that the solution it has will always work.

```
function SIMPLE-PROBLEM-SOLVING-AGENT(percept) returns an action
   persistent: seq, an action sequence, initially empty
               state, some description of the current world state
               goal, a goal, initially null
               problem, a problem formulation

   state ← UPDATE-STATE(state, percept)
   if seq is empty then
      goal ← FORMULATE-GOAL(state)
      problem ← FORMULATE-PROBLEM(state, goal)
      seq ← SEARCH(problem)
      if seq = failure then return a null action
   action ← FIRST(seq)
   seq ← REST(seq)
   return action
```

A simple problem-solving agent. It first formulates a goal and a problem, searches for a sequence of actions that would solve the problem, and then executes the actions one at a time. When this is complete, it formulates another goal and starts over.

1.16 Problem formulation(Problem Definition)

A problem can be defined formally by five components:
1. Initial state ,the agent starts in.
2. Actions: A description of the possible actions available to the agentGiven a particular state s, ACTIONS (s) returns a set of actions that can be executed in s.We say that each of these actions is applicable in s.
3. Transition Model: A description of what each action does.Transition model, specified by a function RESULT(s, a) that returns the state that results from doing action a in state s. Also use the term successor to refer to any state reachable from a given state by a single action.
STATE SPACE: Together, the initial state, actions, and transition model define the state space of the problem that is the set of all states reachable from the initial state by any sequence of actions.The state space forms a directed network or graph in which the nodes are states and the links between nodes are actions. A path in the state space is a sequence of states connected by a sequence of actions.
4. Goal Test, determines whether a given state is a goal state
Explicit: Among the set of possible goal state(s), check whether the given state is one of them
implicit: Goal is specified by an abstract property , for ex. in chess, the goal is to reach a state called "checkmate," where the opponent's king is under attack and can't escape.
5. Path cost : A function that assigns a numeric cost to each path. PSA should choose a cost function that reflects its own performance measure. The step cost of taking action a to go from state s to state s' is denoted as c (s, a, s') , assumed to be ≥ 0

Five elements define a problem and can be represented as a single data structure that is given as input to a problem solving algorithm. A solution to a problem is a path (sequence of actions) from the initial state to a goal state. Solution quality is measured by the path cost function. An optimal solution has the lowest path cost among all solutions.

EXAMPLE PROBLEMS

A TOY PROBLEM is intended to illustrate or exercise various problem-solving methods. It can be given a concise, exact description and hence is usable by different researchers to compare the performance of algorithms

A REAL-WORLD PROBLEM is one whose solutions people actually care about. It cannot have a single agreed-upon description, but we can give the general description.

Vacuum cleaning world This can be formulated as a problem as follows:

• States: The state is determined by both the agent location and the dirt locations. The agent is in one of two locations, each of which might or might not contain dirt. Thus, there are 2 × 2^2 = 8 possible world states. A larger environment with n locations has n*2^n states.

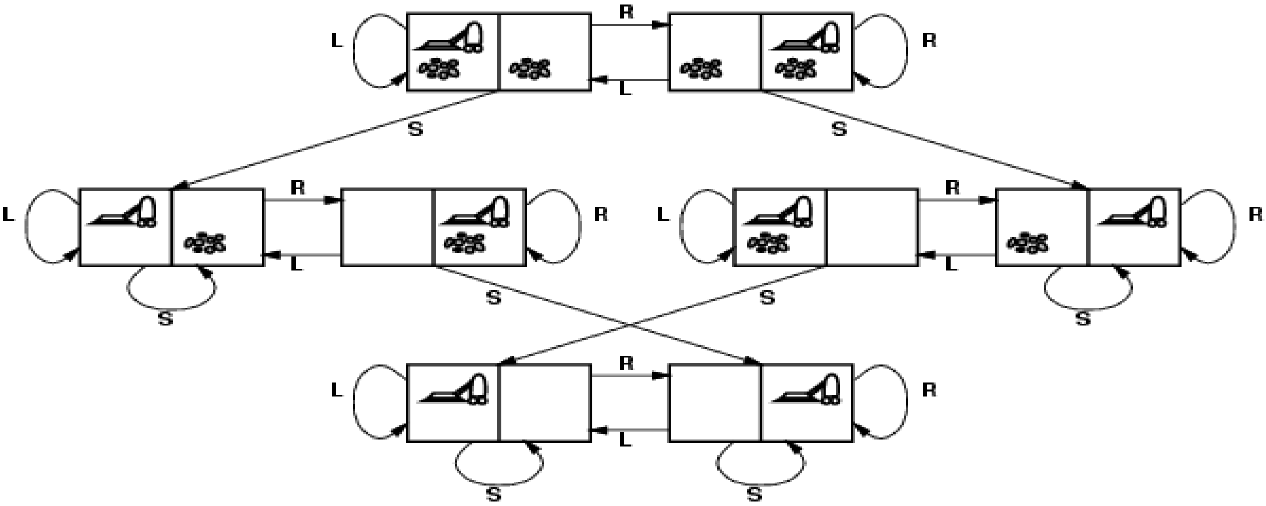
• Initial state: Any state can be designated as the initial state.

• Actions: In this simple environment, each state has just three actions: Left, Right, and Suck. Larger environments might also include Up and Down.

• Transition model: The actions have their expected effects, except that moving Left in the leftmost square, moving Right in the rightmost square, and Sucking in a clean square have no effect. • Goal test: This checks whether all the squares are clean.

• Path cost: Each step costs 1, so the path cost is the number of steps in the path.

Vacuum world state space graph – Summary



Arcs denote actions { L, R, S}

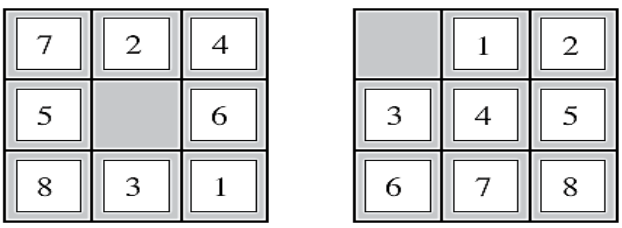
States: dirt and agent location

Actions: Left, Right, Suck

goal test: no dirt at all locations

path cost: 1 per action

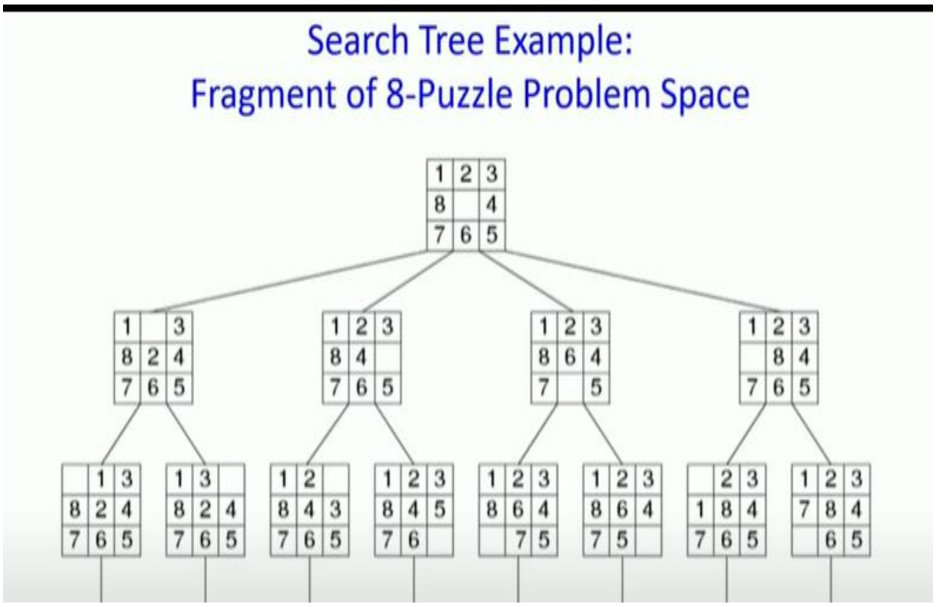
8-puzzle: consists of a 3×3 board with eight numbered tiles and a blank space. A tile adjacent to the blank space can slide into the space.



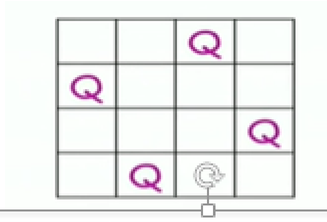
Start State Goal State

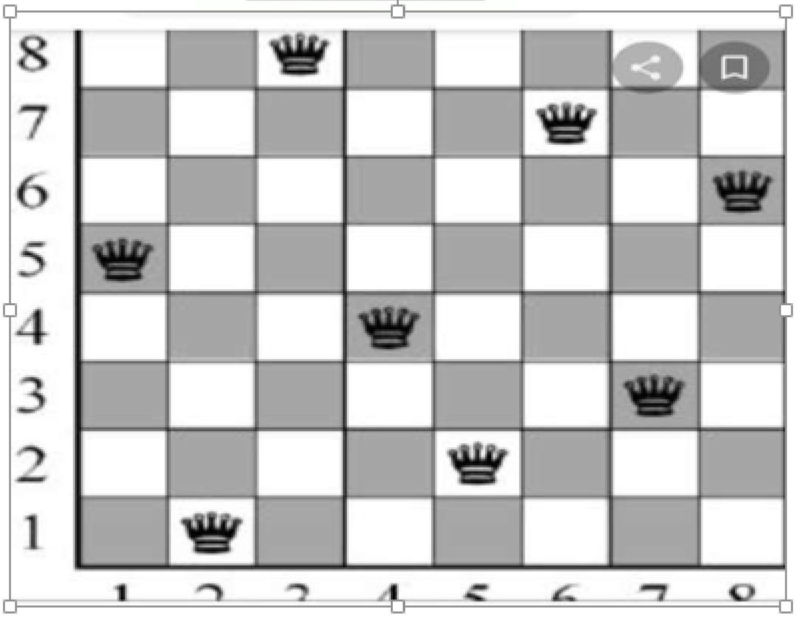
States: A state description specifies the location of each of the eight tiles and the blank in one of the

nine squares.
Initial state: Any state can be designated as the initial state.
Actions: The simplest formulation defines the actions as movements of the blank space Left, Right, Up, or Down.
Transition model: Given a state and action, this returns the resulting state;
for example, if we apply Left to the start state in Figure, the resulting state has the 5 and the blank switched. Goal test: This checks whether the state matches the goal configuration shown in Figure Path cost: Each step costs 1, so the path cost is the number of steps in the path.



8-queens problem: The goal of the 8-queens problem is to place eight queens on a chessboard such that no queen attacks any other. A queen attacks any piece in the same row, col or diagonal Efficient algorithms exist for whole n-queens familyStill, an interesting test problem for search algorithms
There are two main kinds of formulation
Incremental formulation: Starting with an empty state For 8-queen problem, each action adds a queen to the state.
Complete state formulation: Starts with all 8 queens on the board Moves them around
In both cases, no path cost because only the final state counts!
The first incremental formulation one might try is the following:
• States: Any arrangement of 0 to 8 queens on the board is a state.
• Initial state: No queens on the board.
• Actions: Add a queen to any empty square.
• Transition model: Returns the board with a queen added to the specified square.
• Goal test: 8 queens are on the board, none attacked.
We have 1.8*10^14 possible sequences to investigate.





Real-world Problems

Route finding
Touring problems (TSP)
VLSI layout
Robot Navigation
Automatic assembly sequencing
Protein design

Route-finding

The airline travel problems solved by a travel-planning Web site:
States: Each state includes a location (e.g., an airport) and the current time. Since the cost of an action (a flight segment) may depend on previous segments, their fare bases, and their status as domestic or international, the state must record extra information about "historical" aspects. Initial state: This is specified by the user's query.Actions: any flight from the current location, in a given seat class, leaving after the current time, leaving enough time for within-airport transfer if needed.
Transition model: The state resulting from taking a flight will have the flight's destination as the current location and the flight's arrival time as the current time.
Goal test: Are we at the final destination specified by the user?
Path cost: This depends on monetary cost, waiting time, flight time, customs and immigration procedures, seat quality, time of day, type of airplane, frequent-flyer mileage awards, etc.
Touring problems are closely related to route-finding problems, but with an important difference. The starting city and destination city is the same and has to visit every city at least once.Traveling sales person problem (TSP) is a touring problem in which each city must be visited exactly once. The aim is to find the shortest tour.
A VLSI layout problem requires positioning millions of components and connections on a chip to minimize area, minimize circuit delays, minimize stray capacitances, and maximize manufacturing yield. usually split into two parts: cell layout and channel routing.
In cell layout, the primitive components of the circuit are grouped into cells, each of which performs

some recognized function. The aim is to place the cells on the chip so that they do not overlap and so that there is room for the connecting wires to be placed between the cells. Channel routing finds a specific route for each wire through the gaps between the cells. These search problems are extremely complex.

Robot navigation is a generalization of the route-finding problem Rather than following a discrete set of routes, a robot can move in a continuous space with (in principle) an infinite set of possible actions and states. For a circular robot moving on a flat surface, the space is essentially two-dimensional. When the robot has arms and legs or wheels that must also be controlled, the search space becomes many dimensional.

Automatic assembly sequencing of complex objects by a robot is to find an order in which to assemble the parts of some object. If the wrong order is chosen, there will be no way to add some part later in the sequence without undoing some of the work already done. Checking a step in the sequence for feasibility is a difficult geometrical search problem closely related to robot navigation. Thus, the generation of legal actions is the expensive part of assembly sequencing. Any practical algorithm must avoid exploring all but a tiny fraction of the state space. Another important assembly problem is protein design, in which the goal is to find a sequence of amino acids that will fold into a three-dimensional protein with the right properties to cure some disease.

Case Study

AI-Based Intelligent Traffic Signal Control System for Smart Cities

Rapid urbanization has resulted in increased traffic congestion, longer travel times, fuel wastage, and higher pollution levels. Conventional traffic signals operate using fixed timers, which cannot adapt to changing traffic conditions.

To address this issue, a Smart City introduces an AI-based Intelligent Traffic Management System (ITMS) that dynamically adjusts signal timings based on real-time traffic data collected from cameras, sensors, GPS devices, and emergency vehicle tracking systems.

Question Bank:

PART-A (Short Answer Questions)

Q.No	Question	CO	PO	Bloom's Level
1	Define Artificial Intelligence. List any four characteristics of AI.	CO1	PO1	L1 – Remember
2	Explain the foundations of Artificial Intelligence.	CO1	PO1	L2 – Understand
3	Discuss the major milestones in the history of AI.	CO1	PO1	L2 – Understand
4	Define an intelligent agent. Give two real-world examples.	CO1	PO1	L1 – Remember
5	What is a rational agent? How does it differ from an intelligent agent?	CO1	PO2	L2 – Understand
6	Explain the PEAS framework with a suitable example.	CO1	PO2	L2 – Understand
7	Differentiate between deterministic and stochastic environments.	CO1	PO2	L2 – Understand
8	Explain the structure of an intelligent agent.	CO1	PO1	L2 – Understand
9	Define problem formulation. List its components.	CO1	PO2	L1 – Remember
10	Explain the role of sensors and actuators in intelligent agents.	CO1	PO1	L2 – Understand

PART-B (Long Answer Questions)

Q.No	Question	CO	PO	Bloom's Level
1	Explain the concept of Artificial Intelligence. Discuss the foundations, evolution, and applications of AI with suitable examples.	CO1	PO1	L2 – Understand
2	Describe the architecture of an intelligent agent. Explain various task environments with neat diagrams and examples.	CO1	PO1, PO2	L3 – Apply
3	Explain the concept of rationality in intelligent agents. Discuss the factors affecting rational behavior with suitable examples.	CO1	PO2	L3 – Apply

4	Develop the PEAS framework for (i) Autonomous Car (ii) Medical Diagnosis System (iii) Smart Vacuum Cleaner.	CO1	PO2, PO3	L3 – Apply
5	Compare Simple Reflex, Model-Based, Goal-Based, Utility-Based, and Learning Agents with neat diagrams.	CO1	PO2	L4 – Analyze
6	Explain problem-solving agents in detail. Discuss the process of problem formulation using a suitable AI problem.	CO1	PO2, PO3	L3 – Apply
7	Compare different types of task environments (Fully Observable vs Partially Observable, Deterministic vs Stochastic, Episodic vs Sequential, Static vs Dynamic, Discrete vs Continuous, Single-Agent vs Multi-Agent).	CO1	PO2	L4 – Analyze
8	**Case Study:** A Smart Traffic Management System is designed to reduce congestion and provide priority to emergency vehicles. (a) Identify the AI problem. (b) Develop the PEAS framework. (c) Classify the environment. (d) Explain rational agent behavior. (e) Formulate the problem with initial state, actions, goal test, and path cost.	CO1	PO2, PO3	L5 – Evaluate
9	**Case Study:** Design an AI-based Patient Monitoring System for hospitals. Explain the intelligent agent architecture, task environment, rationality, and problem formulation. Evaluate the effectiveness of the proposed solution.	CO1	PO2, PO3	L5 – Evaluate
10	**Case Study:** Design an intelligent warehouse robot for an e-commerce company. Specify its sensors, actuators, PEAS description, environment characteristics, rational decision-making, and problem formulation. Justify your design choices.	CO1	PO2, PO3	L6 – Create

TEXT BOOKS:

3. S. Russel and P. Norvig, "Artificial Intelligence – A Modern Approach", Second Edition, Pearson Education, 2007.
4. Kevin Night and Elaine Rich, Nair B., "Artificial Intelligence (SIE)", Mc Graw Hill, 2008.

REFERENCE BOOKS:

5. David Poole, Alan Mackworth, Randy Goebel, "Computational Intelligence: a logical approach", Oxford University Press, 1998.
6. G. Luger, "Artificial Intelligence: Structures and Strategies for complex problem solving", Sixth Edition, Pearson Education, 2009.
7. J. Nilsson, "Artificial Intelligence: A new Synthesis", 1st Edition, Elsevier Publishers, 1997.
8. Saroj Kaushik, "Artificial Intelligence", 2nd Edition, CENGAGE Learning, 2022.

REFERENCE WEBSITE:

3. https://ai.google/
4. https://swayam.gov.in/nd1_noc19_me71/preview