SYLLABUS

UG - BCA/COMPUTER APPLICATION PAPERS (DSC16)

ARTIFICIAL INTELLIGENCE AND APPLICATIONS

CONTENTS	HOURS (52)
UNIT – 1	
Introduction: What is Artificial Intelligence?, Foundations of AI, History, AI - Past, Present and Future. Intelligent Agents- Environments- Specifying the Task Environment, Properties of Task Environments. **Agent Based Programs:** Structure of Agents, Types of Agents-Simple Reflex Agents, Model-based Reflex Agents, Goal-based Agents; and Utility-based Agents.	10
UNIT – 2	
Problem Solving by Searching: Problem-solving Agents, Well-defined Problems and Solutions, Examples Problems, Searching for Solutions, Uninformed Search Strategies-Breadth-first Search, Uniform-cost Search, Depth-first Search, Depth-limited Search, Iterative Deepening Depth-first Search, Bidirectional Search, Greedy Best-first Search, A* Search, AO* Search Informed (Heuristic) Search Strategies, Heuristic Functions.	10
UNIT – 3	
Knowledge Representation: Knowledge-based Agents, The Wumpus World, Logic, Propositional Logic, Propositional Theorem Proving, Effective Propositional Model Checking, Agents-based on Propositional Logic, First-order Logic-syntax and Semantics of First-order Logic, Using First-order Logic, Unification and Lifting Forward Chaining, Backward Chaining.	12
UNIT – 4	
Learning: Forms of Learning, Supervised Learning, Machine Learning – Decision Trees, Regression and Classification with Linear Models, Artificial Neural Networks, Support Vector Machines.	10
UNIT – 5	
Applications of AI: Natural Language Processing, Text Classification and Information Retrieval, Speech Recognition, Image Processing and Computer Vision, Robotics.	10

CONTENTS

UNIT 1

ARTIFICIAL INTELLIGENCE AND INTELLIGENT AGENTS

1. INTRODUCTION

In which we try to explain why we consider artificial intelligence to be a subject most worthy of study, and in which we try to decide what exactly it is, this being a good thing to decide before embarking.

Intelligence: We call ourselves *Homo sapiens*—man the wise—because our **intelligence** is so important to us. For thousands of years, we have tried to understand *how we think*; that is, how a mere handful of matter can perceive, understand, predict, and manipulate a world far larger and more complicated than itself.

Artificial Intelligence: The field of **artificial intelligence**, or AI, goes further still: it attempts not just to understand but also to *build* intelligent entities.

Artificial Intelligence is one of the newest fields in science and engineering. Work started in earnest soon after World War II, and the name itself was coined in 1956. Along with molecular biology, AI is regularly cited as the "field I would most like to be in" by scientists in other disciplines. A student in physics might reasonably feel that all the good ideas have already been taken by Galileo, Newton, Einstein, and the rest. AI, on the other hand, still has openings for several full-time Einstein's and Edison's.

AI currently encompasses a huge variety of subfields, ranging from the general (learning and perception) to the specific, such as playing chess, proving mathematical theorems, writing poetry, driving a car on a crowded street, and diagnosing diseases. AI is relevant to any intellectual task; it is truly a universal field.

1.1 WHAT IS ARTIFICIAL INTELLIGENCE?

We have claimed that AI is exciting, but we have not said what it is. Below given eight definitions of artificial intelligence, laid out along two main dimensions namely, 1. Thinking versus Acting. 2. Human performance versus Rationality.

Historically, all four approaches to AI have been followed.

1. ***Thinking Humanly***

"The exciting new effort to make computer 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)

A human-centered approach must be in part an empirical science, involving observation and hypotheses about human behaviour. A rationalist approach involves a combination of mathematics and engineering.

Let us look at the four approaches in more detail.

1.1.1 Acting Humanly: The Turing Test Approach

The Turing Test, proposed by Alan Turing (1950), was designed to provide a satisfactory operational definition of intelligence. A computer passes the test if a human interrogator, after posing some written questions, cannot tell whether the written responses come from a person or from a computer.

For now, we note that programming a computer to pass a rigorously applied test provides plenty to work on. The computer would need to possess 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.

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, the so-called ***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." To pass the total Turing Test, the computer will need

- **COMPUTER VISION** to perceive objects, and
- **ROBOTICS** to manipulate objects and move about.

1.1.2 Thinking Humanly: The Cognitive Modeling Approach

If we are going 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* the actual workings of human minds. There are three ways to do this:

Through Introspection – trying to catch our own thoughts as they go by;

Through Psychological Experiments – observing a person in action; and

Through Brain Imaging – observing the brain in action.

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 corresponding human behavior, that is evidence that some of the program's mechanisms could also be operating in humans.

For example, Allen Newell and Herbert Simon, who developed GPS, the ***"General Problem Solver"*** (Newell and Simon, 1961), were not content merely to have their program solve problems correctly. They were more concerned with comparing the trace of its 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 human mind.

Cognitive Science is a fascinating field in itself, worthy of several textbooks and at least one encyclopedia (Wilson and Keil, 1999). We will occasionally comment on similarities or differences between AI techniques and human cognition. Real cognitive science, however, is necessarily based on experimental investigation of actual humans or animals. We will leave that for other books, as we assume the reader has only a computer for experimentation.

In the early days of AI there was often confusion between the approaches: an author would argue that an algorithm performs well on a task and that it is therefore a good model of human performance, or vice versa.

Modern authors separate the two kinds of claims; this distinction has allowed both AI and cognitive science to develop more rapidly. The two fields continue to fertilize each other, most notably in computer vision, which incorporates neurophysiological evidence into computational models.

1.1.3 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 reasoning processes. His syllogisms 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 the field called ***logic.***

Logicians in the 19th century developed a precise notation for statements about all kinds of objects in the world and the relations among them. (Contrast this with ordinary arithmetic notation, which provides only for statements about numbers.) By 1965, programs existed that could, in principle, *solve any solvable problem described in logical notation.* The so-called ***logicist tradition*** within artificial intelligence hopes to build on such programs to create intelligent systems.

There are two main obstacles to this approach. *First,* it is not easy to take informal knowledge and state it in the formal terms required by logical notation, particularly when the knowledge is less than 100% certain. *Second,* there is a big difference between solving a problem ***"in principle"*** and solving it in practice. Even problems with just a few tons' 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.*

1.1.4 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 expected to do more: operate

autonomously, perceive their environment, persist over a prolonged time period, adapt to change, and create and pursue goals. A ***rational agent*** is one that acts so as to achieve the best outcome or, when there is uncertainty, the best expected outcome.

In the ***"laws of thought"*** approach to AI, the 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. On the other hand, correct inference is not all of rationality; in some situations, there is no provably correct thing to do, but something must still be done. There are also ways of acting rationally that cannot be said to involve inference.

The rational-agent approach has two advantages over the other approaches. First, it is more general than *the "laws of thought" approach* because correct inference is just one of several possible mechanisms for achieving rationality. Second, it is more amenable to scientific development than are approaches based on human behavior or human thought.

1.2 FOUNDATIONS OF AI

In this section, we provide a brief history of the disciplines that contributed ideas, viewpoints and techniques to Artificial Intelligence.

1.2.1 Philosophy (The Study of the Fundamental Nature of Knowledge)

- ❖□ Can formal rules be used to draw valid conclusions?
- ❖□ How does the mind arise from a physical brain?
- ❖□ Where does knowledge come from?
- ❖□ How does knowledge lead to action?

➢ **Aristotle (384–322 B.C.),** was the first to formulate a precise set of laws governing the rational part of the mind. He developed an informal system of syllogisms for proper reasoning, which in principle allowed one to generate conclusions mechanically, given initial premises.

Example:

All dogs are animals;

All animals have four legs;

Therefore, all dogs have four legs.

➢ **Ramon Lull (d. 1315)** had the idea that useful reasoning could actually be carried out by a mechanical artifact.

- **Thomas Hobbes (1588–1679)** proposed that reasoning was like numerical computation that *"we add and subtract in our silent thoughts"*.
- The automation of computation itself was already well under way. Around 1500, **Leonardo da Vinci (1452–1519)** designed but did not build a mechanical calculator; recent reconstructions have shown the design to be functional.
- The first known calculating machine was constructed around 1623 by the German scientist **Wilhelm Schickard (1592–1635)**
- The Pascaline, built-in 1642 by **Blaise Pascal (1623–1662)** wrote that "the arithmetical machine produces effects which appear nearer to thought than all the actions of animals."
- **Gottfried Wilhelm Leibniz (1646–1716)** built a mechanical device intended to carry out operations on concepts rather than numbers, but its scope was rather limited.
- **Rene Descartes (1596–1650)** gave the first clear discussion of the distinction between mind and matter and of the problems that arise.
- The empiricism movement, starting with **Francis Bacon's (1561–1626)** Novum Organum, is characterized by a dictum of **John Locke (1632–1704):** "Nothing is in the understanding, which was not first in the senses."
- **David Hume's (1711–1776)** *A Treatise of Human Nature* (Hume, 1739) proposed what is now known as *the principle of induction*: that general rules are acquired by exposure to repeated associations between their elements.
- The confirmation theory of **Carnap and Carl Hempel (1905-1997)** attempted to analyze the acquisition of knowledge from experience.

  Carnap's book: The Logical Structure of the World (1928) defined an explicit computational procedure for extracting knowledge from elementary experiences. It was probably the first theory of mind as a computational process.
- The final element in the philosophical picture of the mind is the connection between knowledge and action. This question is vital to AI because intelligence requires action as well as reasoning. Moreover, only by understanding how actions are justified can we understand how to build an agent whose actions are justifiable (or rational).

1.2.2 Mathematics

❖□ What are the formal rules to draw valid conclusions?

❖□ What can be computed?

❖□ How do we reason with uncertain information?

Formal science required a level of mathematical formalization in three fundamentals areas: **logic, computation, and probability**.

Logic

Earl Stanhope's Logic Demonstrator was a machine that was able to solve syllogisms, numerical problems in a logical form, and elementary questions of probability.

- ***George Boole (1815–1864),*** who worked out the details of propositional, or Boolean, logic (Boole, 1847).
- In 1879, ***Gottlob Frege (1848–1925)*** extended Boole's logic to include objects and relations, creating the first order logic that is used today.

  First order logic – Contains predicates, quantifiers and variables

  Ex.: Philosopher(a) ⇒ Scholar(a)

  ∀x, effect_carona(x) ⇒ quarantine(x)

  ∀x, King(x) ^ Greedy (x) ⇒ Evil (x)
- ***Alfred Tarski (1902–1983)*** introduced a theory of reference that shows how to relate the objects in a logic to objects in the real world.

Logic and Computation: The first nontrivial algorithm is thought to be Euclid's algorithm for computing greatest common divisors (GCD).

- Beside logic and computation, the third great contribution of mathematics to AI is the probability. The Italian **Gerolamo Cardanao (1501–1576)** first framed the idea of probability, describing it in terms of the possible outcomes of gambling events.
- **Thomas Bayes (1702–1761)** proposed a rule for updating probabilities in the light of new evidence. Baye's rule underlies most modern approaches to uncertain reasoning in AI systems.

1.2.3 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?

The science of economics got its start in 1776, when Scottish philosopher Adam Smith (1723–1790) treat it as a science, using the idea that economies can be thought of as consisting of individual agents maximizing their own economic well-being.

Decision theory, which combines probability theory with utility theory, provides a formal and complete framework for decisions (economic or otherwise) made under uncertainty–that is, in cases where probabilistic descriptions appropriately capture the decision maker's environment. This is suitable for "large" economies where each agent need pay no attention to the actions of other agents as individuals. For "small" economies, the situation is much more like a game: the actions of one player can significantly affect the utility of another (either positively or negatively).

Von Neumann and Morgenstern's development of ***game theory*** included the surprising result that, for some games, a rational agent should adopt policies that are randomized. Unlike decision theory, game theory does not offer an unambiguous prescription for selecting actions.

The work of ***Richard Bellman (1957)*** formalized a class of sequential decision problems called Markov decision processes. Work in economics and operations research has contributed much to our notion of rational agents, yet for many years AI research developed along entirely separate paths.

One reason was the apparent complexity of making rational decisions. The pioneering AI researcher ***Herbert Simon (1916–2001)*** won the Nobel Prize in economics in 1978 for his early work showing that models based on satisficing–making decisions that are "good enough,"

1.2.4 Neuroscience

❖□ How do brain process information?

Neuroscience is the study of the nervous system, particularly the brain.

➢ 335 B.C. Aristotle wrote, "Of all the animals, man has the largest brain in proportion to his size."

➢ Paul Broca's (1824–1880) study of aphasia (speech deficit) in brain-damaged patients in 1861 demonstrated the existence of localized areas of the brain responsible for specific cognitive functions. In particular, he showed that speech production was localized to the portion of the left hemisphere now called Broca's area. By that time, it was known that NEURON the brain consisted of nerve cells, or neurons, but it was not until 1873 that Camillo Golgi (1843–1926) developed a staining technique allowing the observation of individual neurons in the brain (Figure 1.1).

➢ This technique was used by Santiago Ramon y Cajal (1852– 1934) in his pioneering studies of the brain's neuronal structures.

➢ Nicolas Rashevsky (1936, 1938) was the first to apply mathematical models to the study of the nervous system.

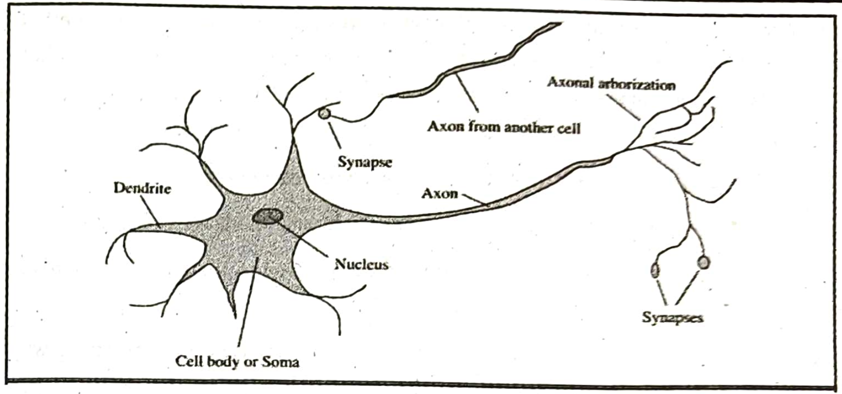

Figure 1.1: A neuron cell of human brain

We now have some data on the mapping between areas of the brain and the parts of the body that they control or from which they receive sensory input. Such mappings are able to change radically over the course of a few weeks, and some animals seem to have multiple maps. Moreover, we do not fully understand how other areas can take over functions when one area is damaged. There is almost no theory on how an individual memory is stored.

➢ The measurement of intact brain activity began in 1929 with the invention by ***Hans Berger*** of the electroencephalograph (EEG).

➢ The recent development of functional magnetic resonance imaging (fMRI) (***Ogawa et al., 1990; Cabeza and Nyberg, 2001***) is giving neuroscientists unprecedentedly detailed images of brain activity, enabling measurements that correspond in interesting ways to ongoing cognitive processes.

➢ Individual neurons can be stimulated electrically, chemically, or even optically ***(Han and Boyden, 2007)***, allowing neuronal input– output relationships to be mapped.

1.2.5 Psychology

❖□ How do humans and animals think and act?

➢ ***Behaviorism movement***, led by ***John Watson (1878-1958)***. Behaviorists insisted on studying only objective measures of the percepts(stimulus) given to an animal and its resulting actions (or response). Behaviorism discovered a lot about rats and pigeons but had less success at understanding human.

➢ ***Cognitive psychology***, which views the brain as an information-processing device, can be traced back at least to the works of ***William James (1842–1910)***.

mind
Conscion

- ***The Nature of Explanation***, by **Bartlett's student and successor Kenneth Craik (1943)**, forcefully re-established the legitimacy of such "mental" terms as beliefs and goals, arguing that they are just as scientific as, say, using pressure and temperature to talk about gases, despite their being made of molecules that have neither.

  Craik specified the three key steps of a ***knowledge-based agent:***

  1. The stimulus must be translated into an internal representation,
  2. The representation is manipulated by cognitive processes to derive new internal representations, and
  3. These are in turn retranslated back into action. He clearly explained why this was a good design for an agent:

- A common view among psychologists that *"a cognitive theory should be like a computer program"* **(Anderson, 1980)**; that is, it should describe a detailed information processing mechanism whereby some cognitive function might be implemented.

1.2.6 Computer Engineering

❖□ How can we build an efficient computer?

For artificial intelligence to succeed, we need two things: ***intelligence and an artifact.*** The computer has been the artifact(object) of choice.

- The first *operational computer* was the electromechanical Heath Robinson, built in 1940 by Alan Turing's team for a single purpose: deciphering German messages.
- The first *operational programmable computer* was the Z-3, the invention of Konrad Zuse in Germany in 1941.
- The first *electronic computer*, the ABC, was assembled by John Atanasoff and his student Clifford Berry between 1940 and 1942 at Iowa State University.
- The first *programmable machine* was a loom, devised in 1805 by Joseph Marie Jacquard (1752-1834) that used punched cards to store instructions for the pattern to be woven.
- In the mid-19th century, Charles Babbage (1792–1871) designed two machines, Analytical Engine was far more ambitious: it included addressable memory, stored programs, and conditional jumps and was the first artifact capable of universal computation.

1.2.7 Control Theory and Cybernetics

❖□ How can artifacts operate under their own control?

- **Ktesibios of Alexandria (c. 250 B.C.)** built the first self-controlling machine: a water clock with a regulator that maintained a constant flow rate. This invention changed the definition of what an artifact could do.
- *Self-regulating feedback control systems* include the steam engine governor, created by James Watt (1736–1819).
- ***Control theory*** was *Norbert Wiener (1894–1964).* Wiener was a brilliant mathematician who worked with Bertrand Russell, among others, before developing an interest in biological and mechanical control systems and their connection to cognition.
- Wiener's Look Cybernetics (1948) became a bestseller and awoke the public to the possibility of artificially intelligent machine.
- Ashby's Design for a Brain (1948, 1952) elaborated on his idea that intelligence could be created by the use of ***homeostatic devices*** containing appropriate feedback loops to achieve stable adaptive behavior.
- *Modern control theory,* especially the branch known as stochastic optimal control, has as its goal the design of systems that maximize an ***objective function*** over time. This roughly matches our view of AI: designing systems that behave optimally.

Calculus and matrix algebra, the tools of control theory, lend themselves to systems that are describable by fixed sets of continuous variables, whereas AI was founded in part as a way to escape from these perceived limitations. The tools of logical inference and computation allowed AI researchers to consider problems such as language, vision, and planning that fell completely outside the control theorist's purview.

1.2.8 Linguistics

❖□ How does language relate to thought?

In 1957, B. F. Skinner published *Verbal Behavior*. This was a comprehensive, detailed account of the behaviorist approach to language learning, written by the foremost expert in the field.

The Linguist Noam Chomsky, who had just published a book on his own theory, *Syntactic Structures*. Chomsky pointed out that the behaviorist theory did not address the notion of creativity in language.

Chomsky's theory–based on *syntactic models* going back to the Indian linguist Panini (c. 350 B.C.) could explain this, and unlike previous theories, it was formal enough that it could in principle be programmed.

Modern linguistics and AI were "born" at about the same time, and grew up together, intersecting in a hybrid field called ***computational linguistics or natural language processing***.

The problem of understanding language soon turned out to be considerably more complex than it seemed in 1957. Understanding language requires an understanding of the subject matter and context, not just an understanding of the structure of sentences.

Knowledge representation (the study of how to put knowledge into a form that a computer can reason with) was tied to language and informed by research in linguistics.

1.3 THE HISTORY OF ARTIFICIAL INTELLIGENCE

Artificial Intelligence is not a new technology for researchers. This technology is much older than you would imagine. Even there are the myths of Mechanical men in Ancient Greek and Egyptian Myths. Following are some milestones in the history of AI which defines the journey from the AI generation to till date development.

1.3.1 The Gestation of Artificial Intelligence (1943–1955)

➢ **In 1943:** The first work that is now generally recognized as AI was done by Warren McCulloch and Walter Pitts (1943). They drew on three sources:

- ✓□ Knowledge of the basic physiology and function of neurons in the brain;
- ✓□ A formal analysis of propositional logic due to Russell and Whitehead; and
- ✓□ Turing's theory of computation.

❖□ **A Model of Artificial Neurons**

- ✓□ Each neuron is characterized as being "on" or "off," with a switch to "on" occurring in response to stimulation by a sufficient number of neighbouring neurons.

❖□ The state of a neuron was conceived of as "factually equivalent to a proposition which proposed its adequate stimulus."

➢ **In 1949:** Donald Hebb (1949) demonstrated a simple updating rule for modifying the connection strengths between neurons. His rule, now called ***Hebbian learning.***

➢ **In 1950:** The Alan Turing who was the London Mathematical Society and pioneered Machine learning in 1950. Alan Turing publishes **"Computing Machinery and Intelligence"** Therein, he introduced ***the Turing Test,*** machine learning, genetic algorithms, and reinforcement learning. He proposed the Child Programme idea, explaining "Instead of trying to produce a programme to simulate the adult mind, why not rather try to produce one which simulated the child's?"

1.3.2 The Birth of Artificial Intelligence (1956)

➢ **In 1955:** An Allen Newell and Herbert A. Simon created the "first artificial intelligence program" which was named as **"Logic Theorist (LT)"**. "We have invented a computer program capable of thinking non-numerically, and thereby solved the venerable mind–body problem."

➢ **In 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- – Computer Science, Mathematics, Physics, and others – with the sole aim of exploring the potential of Synthetic Intelligence.

At that time high-level computer languages such as FORTRAN, LISP, or COBOL were invented and the enthusiasm for AI was very high at that time.

1.3.3 Early Enthusiasm, Great Expectations (1952–1969)

The early years of AI were full of successes- in a limited way. The *intellectual* establishment, by and large, preferred to believe that "a machine can never do X." AI researchers naturally responded by demonstrating one X after another.

Newell and Simon's early success was followed up with the General Problem Solver, or GPS. Unlike Logic Theorist, this program was designed from the start to imitate human problem-solving protocols.

Models of Cognition led Newell and Simon (1976) to formulate the famous physical symbol system hypothesis, which states that "a physical symbol system has the necessary and sufficient means for general intelligent action."

In 1958, McCarthy published a paper entitled Programs with Common Sense, in which he described the Advice Taker, a hypothetical program that can be seen as the first complete AI system.

Hebb's learning methods were enhanced by Bernie Widrow (Widrow and Hoff, 1960; Widrow, 1962), who called his **networks adalines**, and by Frank Rosenblatt (1962) with his **perceptrons.** The ***perceptron convergence theorem*** (Block et al., 1962) says that the learning algorithm can adjust the connection strengths of a perceptron to match any input data, provided such a match exists.

1.3.4 A Dose of Reality (1966–1973)

From the beginning, AI researchers were not shy about making predictions of their coming successes. The following statement by Herbert Simon in 1957 is often quoted:

> It is not my aim to surprise or shock you–but the simplest way I can summarize is to say that there are now in the world machines that think, that learn and that create.

Moreover, their ability to do these things is going to increase rapidly until–in a ***visible future***–the range of problems they can handle will be coextensive with the range to which the human mind has been applied.

In 1966: A report by an advisory committee found that "there has been no machine translation of general scientific text, and none is in immediate prospect."

Early experiments in **machine evolution** (now called **genetic algorithms**) (Friedberg, 1958; Friedberg et al., 1959) were based on the undoubtedly correct belief that by making an appropriate series of small mutations to a machine-code program, one can generate a program with good performance for any particular task. The idea, then, was to try random mutations with a selection process to preserve mutations that seemed useful.

1.3.5 Knowledge-Based Systems: The Key to Power? (1969–1979)

- The picture of problem solving that had arisen during the first decade of AI research was of a general-purpose search mechanism trying to string together elementary reasoning steps to find complete solutions. Such approaches have been called **weak methods** because, although general, they do not scale up to large or difficult problem instances.
- The DENDRAL (Buchanan et al., 1969) researchers consulted analytical chemists and found that they worked by looking for well-known patterns of peaks in the spectrum that suggested common substructures in the molecule.

  For example, the following rule is used to recognize a ketone (C=O) subgroup (which weighs 28):

  if there are two peaks at x1 and x2 such that

  (a) $x1 + x2 = M + 28$ (M is the mass of the whole molecule);

  (b) $x1 - 28$ is a high peak;

  (c) $x2 - 28$ is a high peak;

  (d) At least one of x1 and x2 is high.

  then there is a ketone subgroup

  Recognizing that the molecule contains a particular substructure reduces the number of possible candidates enormously.

- Feigenbaum and others at Stanford began the ***Heuristic Programming Project (HPP)*** to investigate the extent to which the new methodology of *EXPERT SYSTEMS* could be applied to other areas of human expertise.
- The next major effort was in the area of **medical diagnosis**. Feigenbaum, Buchanan, and Dr. Edward Shortliffe developed MYCIN to *diagnose blood infections*.

- Two major differences from DENDRAL:
  - ✓□ First, unlike the DENDRAL rules, *no general theoretical model* existed from which the MYCIN rules could be deduced. They had to be acquired from extensive interviewing of experts, who in turn acquired them from textbooks, other experts, and direct experience of cases.
  - ✓□ Second, the rules had *to reflect the uncertainty associated* with medical knowledge. MYCIN incorporated a calculus of uncertainty called *certainty factors* which seemed (at the time) to fit well with how doctors assessed the impact of evidence on the diagnosis.

1.3.6 AI Becomes an Industry (1980–Present) indent

- The first successful commercial expert system, R1, began operation at the Digital Equipment Corporation (McDermott, 1982).
- In 1981, the Japanese announced the "Fifth Generation" project, a 10-year plan to build intelligent computers running Prolog.
- The United States formed the *Microelectronics and Computer Technology Corporation (MCC)* as a research consortium designed to assure national competitiveness.
- In both cases, AI was part of a broad effort, including chip design and human-interface research.
- Overall, the AI industry boomed from a few million dollars in 1980 to billions of dollars in 1988, including hundreds of companies building expert systems, vision systems, robots, and software and hardware specialized for these purposes.

1.3.7 The Return of Neural Networks (1986–Present)

In the mid-1980s at least four different groups reinvented the back-propagation learning algorithm first found in 1969 by Bryson and Ho.

The algorithm was applied to many **learning problems** in computer science and psychology, and the widespread dissemination of the results in the collection ***Parallel Distributed Processing*** (Rumelhart and McClelland, 1986) caused great excitement.

1.3.8 AI Adopts the Scientific Method (1987–Present)

Recent years have seen a revolution in both the content and the methodology of work in artificial intelligence.

- In recent years, approaches based on ***hidden Markov models*** (HMMs) have come to dominate the area. Two aspects of HMMs are relevant.
  - ✓□ First, they are based on a rigorous mathematical theory.

✓☐ Second, they are generated by a process of training on a large corpus of real speech data.

- In the 1950s there was initial enthusiasm for an approach based on sequences of words, with models learned according to the principles of information theory.
- *Neural networks* also fit this trend. Much of the work on neural nets in the 1980s was done in an attempt to scope out what could be done and to learn how neural nets differ from *"traditional" techniques.*
- Judea Pearl's (1988) ***Probabilistic Reasoning in Intelligent Systems*** led to a new acceptance of probability and decision theory in AI, following a resurgence of interest epitomized by Peter Cheeseman's (1985) article ***"In Defense of Probability."***

1.3.9 The Emergence of Intelligent Agents (1995–Present)

Allen Newell, John Laird, and Paul Rosenbloom on SOAR (Newell, 1990; Laird et al., 1987) is the best-known example of a complete ***agent architecture.***

Artificial General Intelligence or AGI (Goertzel and Pennachin, 2007), which held its first conference and organized the Journal of Artificial General Intelligence in 2008.

1.3.10 The Availability of Very Large Data Sets (2001–Present)

Throughout the 60-year history of computer science, the emphasis has been on the algorithm as the main subject of study, But some recent work in AI suggests that for many problems, it makes more sense to worry about the data and be less picky about what algorithm to apply. This is true because of the increasing availability of very large data sources: for example, trillions of words of English and billions of images from the Web (Kilgarriff and Grefenstette, 2006); or billions of base pairs of genomic sequences (Collins et al., 2003).

"Knowledge bottleneck" in AI–the problem of how to express all the knowledge that a system needs–may be solved in many applications by learning methods rather than hand-coded knowledge engineering, provided the learning algorithms have enough data to go on (Halevy et al., 2009).

Study of

1.4 INTELLIGENT AGENTS

An AI system can be defined as the study of the rational agent and its environment.

1.4.1 Agent and Environments

process (based on reason, decision making)

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 1.2.

upon

✓ **Human agent** has eyes, ears, and other organs for sensors; hands, legs, mouth, and other body parts for actuators.

Ex) Self driving Car - agent
horn, moving (to show obs..., traffic) sensor

✓ **Robotic agent** might have cameras and infrared range finders for sensors; 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.

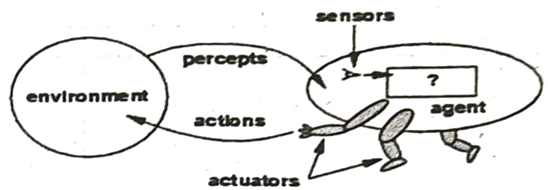

Figure 1.2: Agents interact with environments through sensors and actuators

In general, an agent's choice of action at any given instant can depend on the entire percept sequence observed to date, but not on anything it hasn't perceived. Mathematically speaking, we say that an agent's behavior is described by the **agent function** that maps any given percept sequence to an action:

$$\boxed{F : P^* \longrightarrow A}$$

We can imagine tabulating the agent function that describes any given agent; for most agents, this would be a very large table–infinite, in fact, unless we place a bound on the length of percept sequences we want to consider. Given an agent to experiment with, we can, in principle, construct this table by trying out all possible percept sequences and recording which actions the agent does in response. The table is, of course, an external characterization of the agent. *Internally,* the agent function for an artificial agent will be implemented by ***an agent program.*** It is important to keep these two ideas distinct. *The agent function is an abstract mathematical description; the agent program is a concrete implementation, running within some physical system.*

To illustrate these ideas, we use a very simple example–the vacuum-cleaner world shown in Figure 1.3. This world is so simple that we can describe everything that happens; it's also a made-up world, so we can invent many variations. This particular world has 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, move to the other square. A partial tabulation of this agent function is shown in Figure 1.4 and an agent program that implements it appears in Figure 1.5.

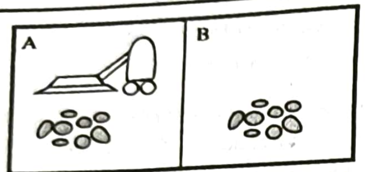

Figure 1.3: A Vacuum-cleaner world with two locations

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*
⋮	⋮
[*A*, *Clean*], [*A*, *Clean*], [*A*, *Clean*]	*Right*
[*A*, *Clean*], [*A*, *Clean*], [*A*, *Dirty*]	*Suck*
⋮	⋮

Figure 1.4: Partial tabulation of a simple agent function for the vacuum-cleaner world

In this figure various vaccuum-world agents can be defined and simply filling in the right-hand side column in various ways.

If the agent uses some randomization to choose its action, then we would have to try each sequence may times to identify the probability of each action.

```
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
```

Figure 1.5: The agent program for a simple reflex agent in the two–state vacuum environment

1.4.1.1 Good Behavior: The Concept of Rationality

A rational agent is one that does the right thing–conceptually speaking, every entry in the table for the agent function is filled out correctly.

Good Behaviour: Rationality → quality being guided by reasons.

Description

A rational agent is an agent that Acts in order to achieve the best outcome, or where there is uncertainty, the best-expected outcome.

Conceptually speaking, it does the "right thing".

When an agent is plunked down 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.

When an agent is plunked down in an environment, it generates a sequence of action 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. This notion of desirability is captured by ***a performance measure*** that evaluates any given sequence of environment states.

useful, attracti

Notice that we said environment states, not agent states. If we define success in terms of agent's opinion of its own performance, an agent could achieve perfect rationality simply by deluding itself that its performance was perfect.

Consider, for the above example, the vacuum-cleaner agent from the preceding section. We might propose to measure performance by the amount of dirt cleaned up in a single eight-hour shift. With a rational agent, of course, what you ask for is what you get. A rational agent can maximize this performance measure by cleaning up the dirt, then dumping it all on the floor, then cleaning it up again, and so on. A more suitable performance measure would reward the agent for having a clean floor.

As a general rule, it is better to design performance measures according to what one actually wants in the environment, rather than according to how one thinks the agent should behave.

1.4.1.2 Rationality

What is rational at any given time depends on four things:

- The performance measure that defines the criterion of success. performance
- The agent's prior knowledge of the environment. Environment PEAS
- The actions that the agent can perform. (Actuator
- The agent's percept sequence to date. (Sensor)

agent → Envr → Action

This leads to a 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 whatever built-in knowledge the agent has.

Consider the simple vacuum-cleaner agent that cleans a square if it is dirty and moves to the other square if not; this is the agent function tabulated in Figure 1.4. Is this a rational agent?

First, we need to say what the performance measure is, what is known about the environment, and what sensors and actuators the agent has. Let us assume the following:

- The *performance measure* awards one point for each clean square at each time step, over a "lifetime" of 1000 times steps.
- *The "geography" of the environment* is known a priori (Figure 1.3) but the dirt distribution and the initial location of the agent are not. Clean squares stay clean and sucking cleans the current square. The Left and Right actions move the agent left and right except when this would take the agent outside the environment, in which case the agent remains where it is.
- The only available actions are *Left, Right, and Suck.*
- The agent correctly perceives its location and whether that location contains dirt.

We claim that *under these circumstances the agent* is indeed rational; its expected performance is at least as high as any other agent's.

State of knowing everything

1.4.1.3 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 is not the same as perfection.
  - ✓☐ Rationality maximizes expected performance, while perfection maximizes actual performance.
  - ✓☐ Our definition of rationality does not require omniscience, then, because the rational choice depends only on the percept sequence to date.
- A rational agent not only to gather information but also **to learn as much as possible** from what it perceives. → conscious, aware
  - ✓☐ The agent's initial configuration could reflect some *prior knowledge* of the environment, but as the agent gains experience this may be modified and augmented.
  - ✓☐ There are extreme cases in which the environment is completely known a priori. In such cases, the agent need not perceive or learn; it simply acts correctly.
- A rational agent should be **autonomous**–it should learn what it can to compensate for partial or incorrect prior knowledge.
- For example, a vacuum-cleaning agent that learns to foresee where and when additional dirt will appear will do better than one that does not.

1.4.2 Specifying the Task Environment

To specify task environment for an AI agent to use the performance measure, the environment, and the agent's actuators and sensors (PEAS).

PEAS description like:

- ✓☐ Performance Measure → How do we assess whether we are doing the right thing?
- ✓☐ Environment → What is the world we are in?
- ✓☐ Actuators → How do we affect the world we are in?
- ✓☐ Sensors → How do we perceive the world we are in?

Consider, example., the task of designing an automated taxi driver:

- o☐ What is the performance measure to which we would like our automated driver to aspire?
- o☐ What is the driving environment that the taxi will face?
- o☐ The actuators for an automated taxi include those available to a human driver: control over the engine through the accelerator and control over steering and braking.
- o☐ The basic sensors for the taxi will include one or more controllable video cameras so that it can see the road; it might augment these with infrared or sonar sensors to detect distances to other cars and obstacles.

Figure 1.6 summarizes the PEAS description for the taxi's task environment. We discuss each element in more detail in the following paragraphs.

Agent Type	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	Cameras, sonar, speedometer, GPS, odometer, accelerometer, engine sensors, keyboard

Figure 1.6: PEAS description for the task environment for an automated taxi

In Figure 1.7, we have sketched the basic PEAS elements for a number of additional agent types.

Agent Type	Performance Measure	Environment	Actuators	Sensors
Medical diagnosis system	Healthy patient, reduced costs	Patient, hospital, staff	Display of questions, tests, diagnoses, treatments, referrals	Keyboard entry of symptoms, findings, patient's answers
Satellite image analysis system	Correct image categorization	Downlink from orbiting satellite	Display of scene categorization	Color pixel arrays
Part-picking robot	Percentage of parts in correct bins	Conveyor belt with parts; bins	Jointed arm and hand	Camera, joint angle sensors
Refinery controller	Purity, yield, safety	Refinery, operators	Valves, pumps, heaters, displays	Temperature, pressure, chemical sensors
Interactive English tutor	Student's score on test	Set of students, testing agency	Display of exercises, suggestions, corrections	Keyboard entry

Figure 1.7: Example of agent types and their PEAS descriptions

1.4.3 Properties of Task Environments

The task environment can be classified based on the following dimensions:

- o☐ Fully vs partially observable — Ex chess — card game
- o☐ Single agent vs multi-agent — Tic tac toe — Ludo
- o☐ Deterministic vs stochastic — depend on move
- o☐ Episodic vs sequential — part picking robot — chess
- o☐ Static vs dynamic — puzzle — Taxi driving
- o☐ Discrete vs continuous
- o☐ Known vs unknown — chess — self-driving car

maze puzzle — foot ball

1. **Fully observable/Partially observable:** In a fully observable environment, the agent can directly observe the complete state of the environment at all points in time.
   - ✔ Fully observable environments are convenient because the agent need not maintain any internal state to keep track of the world.
   - ✔ 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.
   - ✔ 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.
   - ✔ If the agent has no sensors at all then the environment is **unobservable.**

2. **Single agent/Multiple agents:** A single agent is involved in a single-agent environment, while multiple agents are involved in a multi-agent environment.
   - ✔ For example, an agent solving a crossword puzzle by itself is clearly in a ***single-agent environment***, whereas an agent playing chess is in a two agent environment.
   - ✔ For example, in chess, the opponent entity B is trying to maximize its performance measure, which, by the rules of chess, minimizes agent A's performance measure. Thus, chess is a ***competitive multiagent environment***.
   - ✔ In the *taxi-driving environment*, on the other hand, avoiding collisions maximizes the performance measure of all agents, so it is a ***partially cooperative multiagent environment.***

Ex maze puzzle — foot ball

3. **Deterministic/non-deterministic:** In a deterministic environment, the outcome of an action is completely determined by the current state of the environment and the action taken by the agent. In contrast, in a non-deterministic environment, the outcome of an action is not completely determined by the current state of the environment and the action taken by the agent.
   - ✔ Taxi driving is clearly stochastic in this sense, because one can never predict the behavior of traffic exactly; moreover, one's tires blow out and one's engine seizes up without warning.
   - ✔ The vacuum world as we described it is deterministic, but variations can include stochastic elements such as randomly appearing dirt and an unreliable suction mechanism

Ex Tic tac toe — Ludo

4. **Episodic/Non-episodic:** In an episodic environment, the agent's experience is divided into atomic episodes, where each episode is independent of the previous episode. In contrast, in a non-episodic environment, the agent's experience is not divided into atomic episodes .
   - ✔ Many classification tasks are episodic.
   - ✔ In ***sequential environments***, 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.

incident — cardgame video game

5. **Static/Dynamic:** In a static environment, the environment does not change while the agent is deliberating. In contrast, in a dynamic environment can change while the agent is deliberating.
   - ✓ Taxi driving is clearly dynamic: the other cars and the 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.

Puzzle — taxi driving agent

6. **Discrete/Continuous:** In a discrete environment, there is a finite number of distinct states. In contrast, in a continuous environment, there is an infinite number of possible states .
   - ✓ For example, the chess environment has a finite number of distinct states (excluding the clock). 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.)

7. **Known/unknown:** A known environment is one where the rules, state transitions, and reward structure of the environment are completely known to the agent. In contrast, an unknown environment is one where the agent has no knowledge of the rules, state transitions, or reward structure of the environment.
   - ✓ In a known environment, the outcomes (or outcome probabilities if the environment is stochastic) for all actions are given. Obviously, if the environment is unknown, the agent will have to learn how it works in order to make good decisions.
   - ✓ A known environment to be partially observable–for example, in solitaire card games.
   - ✓ An unknown environment can be fully observable–in a new video game.

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

Figure 1.8: Examples of task environments and their characteristics

1.5 THE STRUCTURE OF AGENTS

The structure of agents can be described as:

Agent = Architecture + Agent Program

1. Architecture = the machinery that an agent executes on.
2. Agent Program = an implementation of an agent function.
3. Agent Programs take the current percept as the input and return an action to the actuators.

There are different types of agents, such as Simple Reflex Agents, Model-Based Reflex Agents, Goal-Based Agents, Utility-Based Agents, and Learning Agents, that vary in their complexity and capabilities.

1.5.1 Agent Based Programs

The agent program takes just the current percept as input because nothing more is available from the environment; if the agent's actions need to depend on the entire percept sequence, the agent will have to remember the percepts.

For example, Figure 1.9 shows a rather 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. The table–an example of which is given for the vacuum world in Figure 1.3–represents explicitly the agent function that the agent program embodies. 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.

```
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
```

Figure 1.9: The Table-Driven-Agent program is invoked for each new percept and returns an action each time. It retains the complete percept sequence in memory

It is instructive to consider why the table-driven approach to agent construction is doomed to failure. Let P be the set of possible percepts and let T be the lifetime of the agent (the total number of percepts it will receive). The lookup table will contain $\sum_{t=1}^{T} |P|^t$ entries.

TABLE-DRIVEN-AGENT does do what we want: it implements the desired agent function. The key challenge for AI is to find out how to write programs that, to the extent possible, produce rational behaviour from a smallish program rather than from a vast table.

1.6 TYPES OF AGENTS

There are different types of agents, such as **Simple Reflex Agents, Model-Based Reflex Agents, Goal-Based Agents, Utility-Based Agents,** and **Learning Agents,** that vary in their complexity and capabilities.

These agents can be grouped into five classes based on their degree of perceived intelligence and capability:

1. **Simple Reflex Agents**: These agents take decisions on the basis of the current percepts and ignore the rest of the percept history. They only succeed in the fully observable environment.
2. **Model-based Reflex Agents**: These agents can work in a partially observable environment and track the situation. They have a model, which is knowledge of the world, and based on the model they perform actions.
3. **Goal-Based Agents**: The knowledge of the current state environment is not always sufficient to decide for an agent 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.
4. **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 agents act based not only on goals but also on the best way to achieve the goal.
5. **Learning Agents**: A learning agent in AI is the type of agent that can learn from its past experiences or has learning capabilities. It starts to act with basic knowledge and then is able to act and adapt automatically through learning.

1.6.1 Simple Reflex Agents

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.

✓ For example, the vacuum agent whose agent function is tabulated in Figure 1.3 is a simple reflex agent, because its decision is based only on the current location and on whether that location contains dirt. An agent program for this agent is shown in Figure 1.10.


```
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
```

Figure 1.10: The agent program for a simple reflex agent in the two-state vacuum environment. This program implements the agent function tabulated in Figure 2.3.

✓ In other words, some processing is done on the visual input to establish the condition we call *"The car in front is braking."* Then, this triggers some established connection in the agent program to the action *"initiate braking."* We call such a connection a **condition–action rule**, written as:

if car-in-front-is-braking then initiate-braking.

✓ Humans also have many such connections, some of which are learned responses (as for driving) and some of which are innate reflexes (such as blinking when something approaches the eye).

✓ The program in Figure 1.11 is specific to one particular vacuum environment. A more general and flexible approach is first to build a *general-purpose interpreter* for condition–action rules and then to create rule sets for specific task environments. Figure 1.12 gives the structure of this general program in schematic form, showing how the *condition–action rules* allow the agent to make the connection from percept to action.

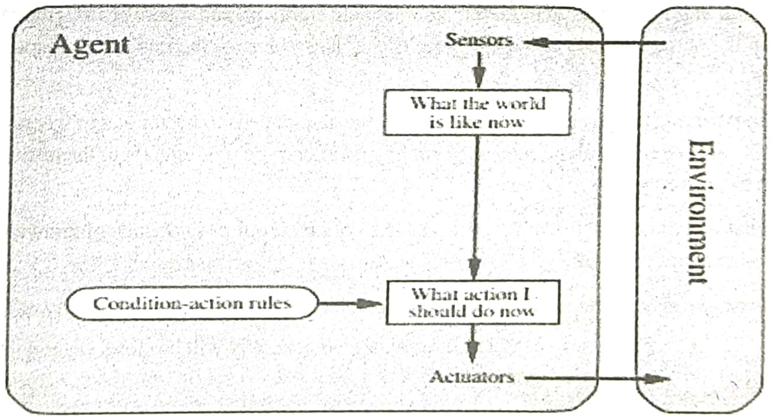

Figure 1.11: Schematic diagram of a simple reflex agent.

```
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
```

Figure 1.12: A simple reflex agent. It acts according to a rule whose condition matches the current state, as defined by the percept.

- ✓ INTERPRET-INPUT function generates an abstracted description of the current state from the percept.
- ✓ RULE-MATCH function returns the first rule in the set of rules that matches the given state description.
- ✓ The agent will work only if the correct decision can be made on the basis of only the current percept–that is, only if the environment is fully observable.
- ✓ Ex: if automated taxi driver is a simple reflex agent - behind a car, it would either brake continuously and unnecessarily or never brake at all.
- ✓ if vacuum cleaner is a simple reflex agent - it would have only dirt sensor and not the location sensor, so only 2 percepts dirty and clean. So, it can't move left or right.

1.6.2 Model-Based Reflex Agents

The most effective way to handle partial observability is for the agent to keep track of the part of the world it can't see now. That is, the agent should maintain some sort of **internal state** that depends on the percept history and thereby reflects at least some of the unobserved aspects of the current state.

For other driving tasks such as changing lanes, the agent needs to keep track of where the other cars are if it can't see them all at once. And for any driving to be possible at all, the agent needs to keep track of where its keys are.

Updating this internal state information as time goes by requires two kinds of knowledge to be encoded in the agent program.

- First, we need some information about how the world evolves independently of the agent.
  - ✓□ For example, that an overtaking car generally will be closer behind than it was a moment ago.
- Second, we need some information about how the agent's own actions affect the world.
  - ✓ For example, that when the agent turns the steering wheel clockwise, the car turns to the right, or that after driving for five minutes northbound on the freeway, one is usually about five miles north of where one was five minutes ago.

This knowledge about "how the world works"–whether implemented in simple Boolean circuits or in complete scientific theories is called a **model** of the world. An agent that uses such a model is called a **model-based agent.**

Figure 1.13 gives the structure of the model-based reflex agent with internal state, showing how the current percept is combined with the old internal state to generate the updated description of the current state, based on the agent's model of how the world works.

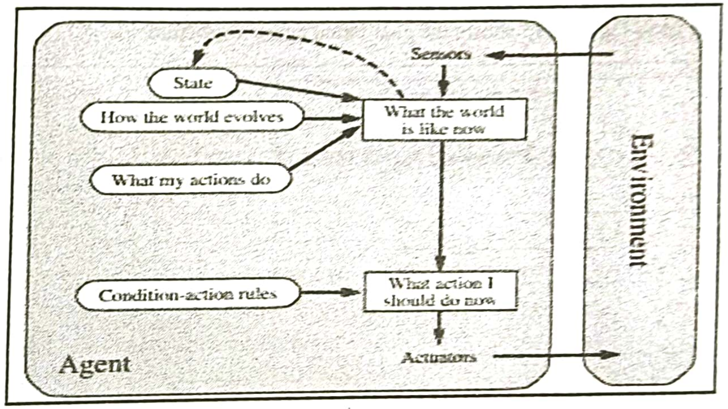

Figure 1.13: A model-based reflex agent

The agent program is shown in Figure 1.14. The interesting part is the function UPDATE-STATE, which is responsible for creating the new internal state description. The details of how models and states are represented vary widely depending on the type of environment and the particular technology used in the agent design.

```
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
```

Figure 1.14: A Model-based reflex agent. It keeps track of the current state of the world, using an internal model. It then chooses an action in the same way as the reflex agent.

Example, the taxi may be driving back home, and it may have a rule telling it to fill up with gas on the way home unless it has at least half a tank. Although "driving back home" may seem to an aspect of the world state, the fact of the taxi's destination is actually an aspect of the agent's internal state.

1.6.3 Goal-Based Agents

Knowing something about the current state of the environment is not always enough to decide what to do. For example, at a road junction, the taxi can turn left, turn right, or go straight on. The correct decision depends on where the taxi is trying to get to. In other words, as well as a current state description, the agent needs some sort of ***goal information*** that describes situations that are desirable–for example, being at the passenger's destination. The agent program can combine this with the model (the same information as was used in the model based reflex agent) to choose actions that achieve the goal. Figure 1.15 shows the goal-based agent's structure.

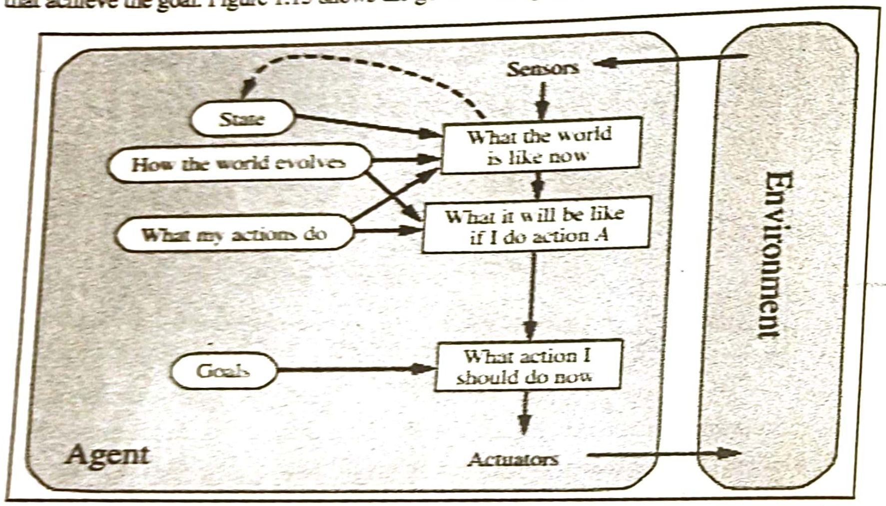

Figure 1.15: A model-based, goal-based agent. It keeps track of the world state as well as a set of goals it is trying to achieve, and chooses an action that will (eventually) lead to the achievement of its goals.

- Goal-Based Action Selection is straightforward-
  - ✓ When goal satisfaction results immediately from a single action.
  - ✓ When the agent has to consider long sequences of twists and turns in order to find a way to achieve the goal.
- ***Search and planning*** are the subfields of AI devoted to finding action sequences that achieve the agent's goals.

1.6.4 Utility-Based Agents

Goals alone are not enough to generate high-quality behavior in most environments. For example, many action sequences will get the taxi to its destination (thereby achieving the goal) but some are quicker, safer, more reliable, or cheaper than others. Goals just provide a crude binary

distinction between ***"happy" and "unhappy"*** states. A more general performance measure should allow a comparison of different world states according to exactly how happy they would make the agent. Because "happy" does not sound very scientific, economists and computer scientists use the term ***utility*** instead.

An agent's ***utility function*** is essentially an internalization of the performance measure. If the internal utility function and the external performance measure are in agreement, then an agent that chooses actions to maximize its utility will be rational according to the external performance measure.

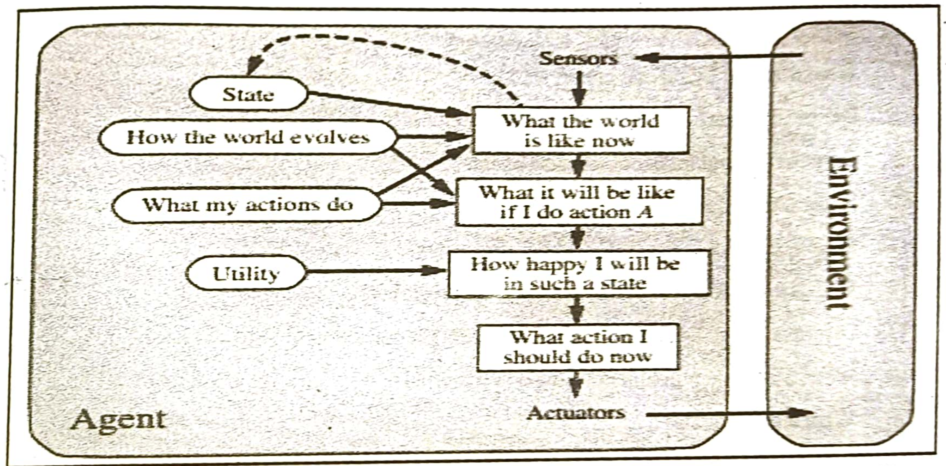

Figure 1.16: A model-based, utility-based agent. It uses a model of the world, along with a utility function that measures its preferences among states of the world. Then it chooses the action that leads to the best expected utility, where expected utility is computed by averaging over all possible outcome states, weighted by the probability of the outcome.

A rational utility-based agent chooses the action that maximizes the expected utility of the action outcomes–that is, the utility the agent expects to derive, on average, given the probabilities and utilities of each outcome.

The utility-based agent structure appears in Figure 1.16. Utility-based agent programs appear in Part IV, where we design decision-making agents that must handle the uncertainty inherent in stochastic or partially observable environments.

A utility-based agent has to model and keep track of its environment, tasks that have involved a great deal of research on perception, representation, reasoning, and learning.

1.7 MODEL QUESTIONS

I. Answer the following questions. (Each question carries TWO marks).

1. Define the intelligent agent.
2. What are the factors that a rational agent should depend on at any given time?
3. List the various types of agent program.
4. Describe the major problems of the simple reflex agent.
5. Mention the basic algorithm for a rational agent.
6. Define AI.
7. What is meant by robotic agent?
8. Define an agent?
9. Give the general model of learning agent.
10. What is the role of agent program?
11. What are PEAS in context to intelligent agents?
12. What are the four different kinds of agent programs?
13. Differentiate an agent function and an agent program.
14. What is a task environment? How it is specified?
15. What are utility based agents?
16. Define the problem solving agent.

II. Answer the following questions. (Each question carries SIX marks).

1. What is an agent? Explain the basic kinds of agent program.
2. What are various characteristics on an agent environment. Describe with example.
3. Write down present and future scope of AI.
4. What do you mean by rational agents? Are the rational agents intelligent? Explain.
5. Distinguish intelligent software agent from intelligent agents in AI.
6. What is a rational agent? Discuss the Automated Taxi driver agent structure with its task environment.
7. What is an intelligent agent? Explain the Vacuum cleaner agent with schematic diagram.
8. Explain a simple reflex agent with a diagram.
9. Write a function for the table driven agent.

III. Answer the following questions. (Each question carries EIGHT marks).

1. Explain properties of environment.
2. Explain in details, the structure of different intelligent agents.
3. Write pseudo code agent programs for the goal-based and utility-based agents.
4. Write down brief history and evolution of AI.
5. Discuss briefly the important properties of task environment with suitable examples.
6. Illustrate the Simple reflex and Model based reflex agent along with examples.
7. Explain with a diagram the goal based reflex agent.
8. Explain with a diagram the model based reflex agent.
9. Explain a simple reflex agent with a diagram.

★★★★★