Artificial Intelligence with Expert System

Unit-I

Overview of Artificial Intelligence

What is AI?

- Artificial Intelligence (AI) is a branch of Science which deals with helping machines find solutions to complex problems in a more human-like fashion.
- This generally involves borrowing characteristics from human intelligence, and applying them as algorithms in a computer friendly way.
- A more or less flexible or efficient approach can be taken depending on the requirements established, which influences how artificial the intelligent behavior appears
- Artificial intelligence can be viewed from a variety of perspectives.

➢ From the perspective of intelligence artificial intelligence is making machines "intelligent" -- acting as we would expect people to act.

- The inability to distinguish computer responses from human responses is called the Turing test.
- Intelligence requires knowledge
- Expert problem solving - restricting domain to allow including significant relevant knowledge

➢ From a business perspective AI is a set of very powerful tools, and methodologies for using those tools to solve business problems.

➢ From a programming perspective, AI includes the study of symbolic programming, problem solving, and search.

- Typically AI programs focus on symbols rather than numeric processing.
- Problem solving - achieve goals.
- Search - seldom access a solution directly. Search may include a variety of techniques.
- AI programming languages include:

– LISP, developed in the 1950s, is the early programming language strongly associated with AI. LISP is a functional programming language with procedural extensions. LISP (LISt Processor) was specifically designed for processing heterogeneous lists -- typically a list of symbols. Features of LISP are run- time type checking, higher order functions (functions that have other functions as parameters), automatic memory management (garbage collection) and an interactive environment.

– The second language strongly associated with AI is PROLOG. PROLOG was developed in the 1970s. PROLOG is based on first order logic. PROLOG is declarative in nature and has facilities for explicitly limiting the search space.

– Object-oriented languages are a class of languages more recently used for AI programming. Important features of object-oriented languages include: concepts of objects and messages, objects bundle data and methods for manipulating the data, sender specifies what is to be done receiver decides how to do it, inheritance (object hierarchy where objects inherit the attributes of the more general class of objects). Examples of object-oriented languages are Smalltalk, Objective C, C++. Object oriented extensions to LISP (CLOS - Common LISP Object System) and PROLOG (L&O - Logic & Objects) are also used.

- Artificial Intelligence is a new electronic machine that stores large amount of information and process it at very high speed
- The computer is interrogated by a human via a teletype It passes if the human cannot tell if there is a computer or human at the other end
- The ability to solve problems
- It is the science and engineering of making intelligent machines, especially intelligent computer programs. It is related to the similar task of using computers to understand human intelligence

Importance of AI

• Game Playing

You can buy machines that can play master level chess for a few hundred dollars. There is some AI in them, but they play well against people mainly through brute force computation--looking at hundreds of thousands of positions. To beat a world champion by brute force and known reliable heuristics requires being able to look at 200 million positions per second.

• Speech Recognition

In the 1990s, computer speech recognition reached a practical level for limited purposes. Thus United Airlines has replaced its keyboard tree for flight information by a system using speech recognition of flight numbers and city names. It is quite convenient. On the other hand, while it is possible to instruct some computers using speech, most users have gone back to the keyboard and the mouse as still more convenient.

• Understanding Natural Language

Just getting a sequence of words into a computer is not enough. Parsing sentences is not enough either. The computer has to be provided with an understanding of the domain the text is about, and this is presently possible only for very limited domains.

• Computer Vision

The world is composed of three-dimensional objects, but the inputs to the human eye and computers' TV cameras are two dimensional. Some useful programs can work solely in two dimensions, but full computer vision requires partial three-dimensional information that is not just a set of two-dimensional views. At present there are only limited ways of representing three-dimensional information directly, and they are not as good as what humans evidently use.

• Expert Systems

A ``knowledge engineer'' interviews experts in a certain domain and tries to embody their knowledge in a computer program for carrying out some task. How well this works depends on whether the intellectual mechanisms required for the task are within the present state of AI. When this turned out not to be so, there were many disappointing results. One of the first expert systems was MYCIN in 1974, which diagnosed bacterial infections of the blood and suggested treatments. It did better than medical students or practicing doctors, provided its limitations were observed. Namely, its ontology included bacteria, symptoms, and treatments and did not include patients, doctors, hospitals, death, recovery, and events occurring in time. Its interactions depended on a single patient being considered. Since the experts consulted by the knowledge engineers knew about patients, doctors, death, recovery, etc., it is clear that the knowledge engineers forced what the experts told them into a predetermined framework. The usefulness of current expert systems depends on their users having common sense.

• Heuristic Classification

One of the most feasible kinds of expert system given the present knowledge of AI is to put some information in one of a fixed set of categories using several sources of information. An example is advising whether to accept a proposed credit card purchase. Information is available about the owner of the credit card, his record of payment and also about the item he is buying and about the establishment from which he is buying it (e.g., about whether there have been previous credit card frauds at this establishment).

The applications of AI

- Consumer Marketing
  - Have you ever used any kind of credit/ATM/store card while shopping?
  - if so, you have very likely been “input” to an AI algorithm
  - All of this information is recorded digitally
  - Companies like Nielsen gather this information weekly and search for
  - patterns
    - ✓ general changes in consumer behavior
    - ✓ tracking responses to new products
    - ✓ identifying customer segments: targeted marketing, e.g., they find out that consumers with sports cars who buy textbooks respond well to offers of new credit cards.
  - Algorithms (“data mining”) search data for patterns based on mathematical theories of learning
- Identification Technologies
  - ID cards e.g., ATM cards
  - can be a nuisance and security risk: cards can be lost, stolen, passwords forgotten, etc
  - Biometric Identification, walk up to a locked door
    - Camera
    - Fingerprint device

    - Microphone
    - Computer uses biometric signature for identification
    - Face, eyes, fingerprints, voice pattern
    - This works by comparing data from person at door with stored library
    - Learning algorithms can learn the matching process by analyzing a large library database off-line, can improve its performance.

- ➢ Intrusion Detection
  - Computer security - we each have specific patterns of computer use times of day, lengths of sessions, command used, sequence of commands, etc
    - would like to learn the “signature” of each authorized user
    - can identify non-authorized users
  - How can the program automatically identify users?
    - record user’s commands and time intervals
    - characterize the patterns for each user
    - model the variability in these patterns
    - classify (online) any new user by similarity to stored patterns
- ➢ Machine Translation
  - Language problems in international business
    - e.g., at a meeting of Japanese, Korean, Vietnamese and Swedish investors, no common language
    - If you are shipping your software manuals to 127 countries, the solution is ; hire translators to translate
    - would be much cheaper if a machine could do this!
  - How hard is automated translation
    - very difficult!
    - e.g., English to Russian
    - not only must the words s be translated, but their meaning also!

AI Applications

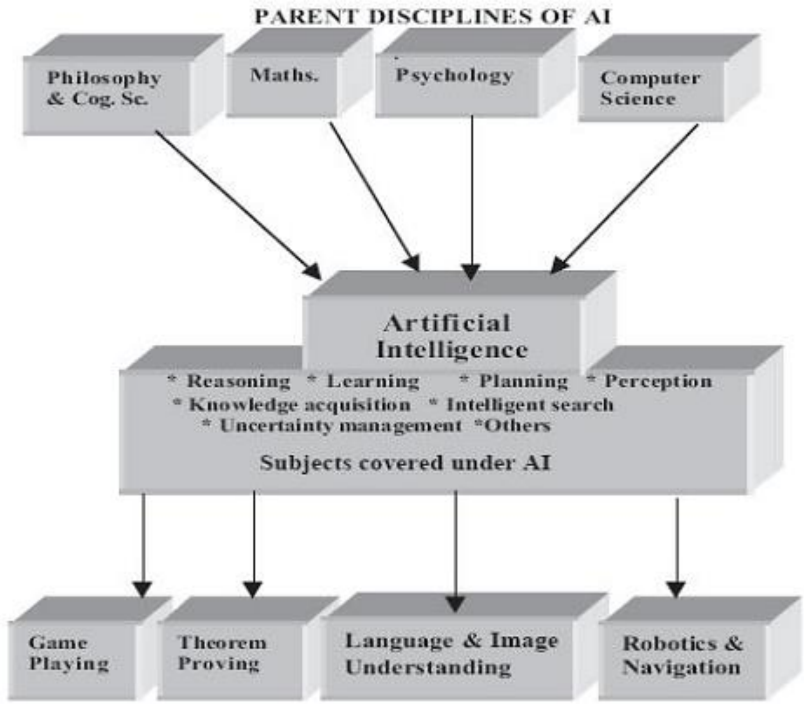


Fig : Application areas of AI

Early work in AI

- "Artificial Intelligence (AI) is the part of computer science concerned with designing intelligent computer systems, that is, systems that exhibit characteristics we associate with intelligence in human behaviour – understanding language, learning, reasoning, solving problems, and so on."
- Scientific Goal To determine which ideas about knowledge representation, learning, rule systems, search, and so on, explain various sorts of real intelligence.
- Engineering Goal To solve real world problems using AI techniques such as knowledge representation, learning, rule systems, search, and so on.
- Traditionally, computer scientists and engineers have been more interested in the engineering goal, while psychologists, philosophers and cognitive scientists have been more interested in the scientific goal.

- The Roots - Artificial Intelligence has identifiable roots in a number of older disciplines, particularly:
    - Philosophy
    - Logic/Mathematics
    - Computation
    - Psychology/Cognitive Science
    - Biology/Neuroscience
    - Evolution
- There is inevitably much overlap, e.g. between philosophy and logic, or between mathematics and computation. By looking at each of these in turn, we can gain a better understanding of their role in AI, and how these underlying disciplines have developed to play that role.
- Philosophy
  - ✓ ~400 BC Socrates asks for an algorithm to distinguish piety from non-piety.
  - ✓ ~350 BC Aristotle formulated different styles of deductive reasoning, which could mechanically generate conclusions from initial premises, e.g. Modus Ponens

    If A ? B and A then B

    If A implies B and A is true then B is true when it's raining you get wet and it's raining then you get wet

  - ✓ 1596 – 1650 Rene Descartes idea of mind-body dualism – part of the mind is exempt from physical laws.
  - ✓ 1646 – 1716 Wilhelm Leibnitz was one of the first to take the materialist position which holds that the mind operates by ordinary physical processes – this has the implication that mental processes can potentially be carried out by machines.
- Logic/Mathematics
  - ✓ ¸ 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.
  - ✓ ¸ 1815 – 1864 George Boole introduced his formal language for making logical inference in 1847 – Boolean algebra.
  - ✓ ¸ 1848 – 1925 Gottlob Frege produced a logic that is essentially the first-order

- ✓ logic that today forms the most basic knowledge representation system.
- ✓ 1906 – 1978 Kurt Gödel showed in 1931 that there are limits to what logic can do. His Incompleteness Theorem showed that in any formal logic powerful enough to describe the properties of natural numbers, there are true statements whose truth cannot be established by any algorithm.
- ✓ 1995 Roger Penrose tries to prove the human mind has non-computable capabilities.

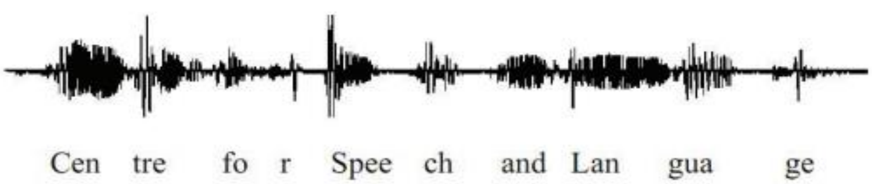
- Computation
  - ✓ 1869 William Jevon's Logic Machine could handle Boolean Algebra and Venn Diagrams, and was able to solve logical problems faster than human beings.
  - ✓ 1912 – 1954 Alan Turing tried to characterise exactly which functions are capable of being computed. Unfortunately it is difficult to give the notion of computation a formal definition. However, the Church-Turing thesis, which states that a Turing machine is capable of computing any computable function, is generally accepted as providing a sufficient definition. Turing also showed that there were some functions which no Turing machine can compute (e.g. Halting Problem).
  - ✓ 1903 – 1957 John von Neumann proposed the von Neuman architecture which allows a description of computation that is independent of the particular realisation of the computer.
  - ✓ 1960s Two important concepts emerged: Intractability (when solution time grows atleast exponentially) and Reduction (to 'easier' problems).
- Psychology / Cognitive Science
  - ✓ Modern Psychology / Cognitive Psychology / Cognitive Science is the science which studies how the mind operates, how we behave, and how our brains process information.
  - ✓ Language is an important part of human intelligence. Much of the early work on knowledge representation was tied to language and informed by research into linguistics.
  - ✓ It is natural for us to try to use our understanding of how human (and other animal) brains lead to intelligent behavior in our quest to build artificial intelligent systems. Conversely, it makes sense to explore the properties of artificial systems(computer models/simulations) to test our hypotheses concerning human systems.

- ✓ Many sub-fields of AI are simultaneously building models of how the human system operates, and artificial systems for solving real world problems, and are allowing useful ideas to transfer between them.

- Biology / Neuroscience
  - ✓ Our brains (which give rise to our intelligence) are made up of tens of billions of neurons, each connected to hundreds or thousands of other neurons.
  - ✓ Each neuron is a simple processing device (e.g. just firing or not firing depending on the total amount of activity feeding into it). However, large networks of neurons are extremely powerful computational devices that can learn how best to operate.
  - ✓ The field of Connectionism or Neural Networks attempts to build artificial systems based on simplified networks of simplified artificial neurons.
  - ✓ The aim is to build powerful AI systems, as well as models of various human abilities.
  - ✓ Neural networks work at a sub-symbolic level, whereas much of conscious human reasoning appears to operate at a symbolic level.
  - ✓ Artificial neural networks perform well at many simple tasks, and provide good models of many human abilities. However, there are many tasks that they are not so good at, and other approaches seem more promising in those areas.
- Evolution
  - ✓ One advantage humans have over current machines/computers is that they have a long evolutionary history.
  - ✓ Charles Darwin (1809 – 1882) is famous for his work on evolution by natural selection. The idea is that fitter individuals will naturally tend to live longer and produce more children, and hence after many generations a population will automatically emerge with good innate properties.
  - ✓ This has resulted in brains that have much structure, or even knowledge, built in at birth.
  - ✓ This gives them at the advantage over simple artificial neural network systems that have to learn everything.
  - ✓ Computers are finally becoming powerful enough that we can simulate evolution and evolve good AI systems.
  - ✓ We can now even evolve systems (e.g. neural networks) so that they are good at learning.

- ✓ A related field called genetic programming has had some success in evolving programs, rather than programming them by hand.

- Sub-fields of Artificial Intelligence
  - ✓ Neural Networks – e.g. brain modelling, time series prediction, classification
  - ✓ Evolutionary Computation – e.g. genetic algorithms, genetic programming
  - ✓ Vision – e.g. object recognition, image understanding
  - ✓ Robotics – e.g. intelligent control, autonomous exploration
  - ✓ Expert Systems – e.g. decision support systems, teaching systems
  - ✓ Speech Processing– e.g. speech recognition and production
  - ✓ Natural Language Processing – e.g. machine translation
  - ✓ Planning – e.g. scheduling, game playing
  - ✓ Machine Learning – e.g. decision tree learning, version space learning
- Speech Processing
  - ✓ As well as trying to understand human systems, there are also numerous real world applications: speech recognition for dictation systems and voice activated control; speech production for automated announcements and computer interfaces.
  - ✓ How do we get from sound waves to text streams and vice-versa?



- Natural Language Processing
  - ✓ For example, machine understanding and translation of simple sentences:
- Planning
  - ✓ Planning refers to the process of choosing/computing the correct sequence of steps to solve a given problem.
  - ✓ To do this we need some convenient representation of the problem domain. We can define states in some formal language, such as a subset of predicate logic, or a series of rules.

  - ✓ A plan can then be seen as a sequence of operations that transform the initial state into the goal state, i.e. the problem solution. Typically we will use some kind of search algorithm to find a good plan.
- Common Techniques
  - ✓ Even apparently radically different AI systems (such as rule based expert systems and neural networks) have many common techniques.
  - ✓ Four important ones are:
    - Knowledge Representation: Knowledge needs to be represented somehow – perhaps as a series of if-then rules, as a frame based system, as a semantic network, or in the connection weights of an artificial neural network.
    - Learning: Automatically building up knowledge from the environment – such as acquiring the rules for a rule based expert system, or determining the appropriate connection weights in an artificial neural network.
    - Rule Systems: These could be explicitly built into an expert system by a knowledge engineer, or implicit in the connection weights learnt by a neural network.
    - Search: This can take many forms – perhaps searching for a sequence of states that leads quickly to a problem solution, or searching for a good set of connection weights for a neural network by minimizing a fitness function.

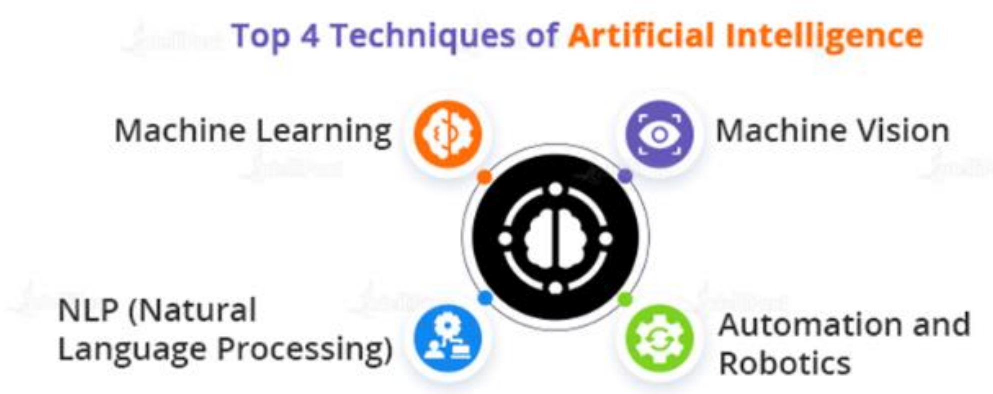
AI Techniques

Depending on the machine's ability to utilize past experiences to anticipate future judgments, memory, and self-awareness, artificial intelligence can be classified into a variety of subcategories.

IBM created the chess program Deep Blue, which can recognize the pieces on the chessboard. However, it lacks the memory necessary to anticipate future behavior. Even if this approach is helpful, it cannot be modified for different circumstances.

The decision-making capabilities of self-driving automobiles serve as an illustration of this type of AI system. Here, observations aid in the quick decisions that must be made since they change often and are not permanently preserved.

The development of technology may also make it feasible to create machines with a sense of awareness that can recognize the status of the world and determine what has to be done. But there are no such systems.



The top four techniques of AI:

Machine Learning:

Machines, in this use of AI, naturally learn from experience rather than being explicitly taught to carry out certain tasks.

Artificial neural networks are the foundation of the subfield of machine learning known as "Deep Learning" that is used for predictive analysis. Unsupervised machine learning, supervised and unsupervised, and reinforcement learning are only a few examples of the many machine learning algorithms.

The algorithm is unsupervised learning and does not employ categorized information to make choices on its own without any direction.

In supervised learning, a feature that includes a combination of an input data set and the intended output is inferred from the training data.

Machines utilize reinforcement learning to determine the best alternative that needs to be considered and to take the appropriate actions to improve the reward.

Machine Vision:

Machines are capable of collecting and analyzing visual data. In this case, cameras are utilized to record sensory information, which is then processed using digital signal processing once the picture is converted from analog to digital.

The data that is produced is then input into a computer. Sensitivity is the capacity of the machine to recognize weak impulses and resolution. The extent to which it can discriminate between objects—are two essential components of machine vision.

Machine vision is used in a variety of applications, including object recognition, medical picture analysis, and signature detection.

Thinking of doing masters in AI? Enroll for a master's in Artificial Intelligence.

NLP(Natural Language Processing)

The way in which computers were trained to comprehend natural languages is via their connections with human language.

Natural Language Processing, the method of extracting meaning from human languages, is trustworthy technology. The machine in NLP records the speech of a person speaking.

Following the audio-to-textual dialogue, the writing is converted to turn the data into audio.

The system then responds to people via audio. Applications of NLP may be found in Interactive Voice Response (IVR) systems used in contact centers, in language translations like Google Translate, and in word processors that verify the correctness of syntax in text, like Microsoft Word.

Be a master in NLP by enrolling in NLP Course in Bangalore.

Automation and Robotics

The goal of automation is to enable machines to perform boring, repetitive jobs, increasing productivity and delivering more effective, efficient, and affordable results. In order to automate processes, many businesses employ machine learning, artificial neural, and graphs.

By leveraging the CAPTCHA technique, this automation can avoid fraud problems during online payments.

Robotic process automation is designed to carry out high-volume, repetitive jobs while being capable of adapting to changing conditions.

Future of Artificial Intelligence

Artificial intelligence (AI) is unquestionably a cutting-edge area of computer science that is poised to dominate a number of burgeoning industries, including data science, robots, and the internet of things. In the upcoming years, this will keep on innovating in the field of technology.

Artificial Intelligence has gone from science fiction to reality in just a few years. Intelligent machines that assist humans exist in real life as well as in science fiction films. We now inhabit a universe of a.i., which was only a tale a few years ago.
Whether we are conscious of it or not, artificial intelligence technology is now engrained in our society and is employed in our daily activities. Nowadays, everyone makes use of AI in their daily lives, from chatbots to Alexa and Siri.

This technology-based industry is advancing and changing. But it wasn't as easy and simple as it seemed to us. It has taken a significant amount of work and the participation of many people to develop AI to this stage.

PROBLEMS, PROBLEM SPACES AND SEARCH

To solve the problem of building a system you should take the following steps:

1. Define the problem accurately including detailed specifications and what constitutes a suitable solution.
2. Scrutinize the problem carefully, for some features may have a central affect on the chosen method of solution.
3. Segregate and represent the background knowledge needed in the solution of the problem.
4. Choose the best solving techniques for the problem to solve a solution.

Problem solving **is a process** of generating solutions from observed data.

- a *'problem'* is characterized by a set of *goals*,
- a set of *objects*, and
- a set of *operations*.

These could be ill-defined and may evolve during problem solving.

- A **'*problem space*'** is an abstract space.
  - ✓ A problem space encompasses all *valid states* that can be generated by the application of any combination of *operators* on any combination of *objects*.
  - ✓ The problem space may contain one or more *solutions*. A solution is a combination of *operations* and *objects* that achieve the *goals*.
- A '***search***' refers to the search for a solution in a problem space.
  - ✓ Search proceeds with different types of '*search control strategies*'.
  - ✓ The *depth-first search and breadth-first search* are the two common *search strategies.*

2.1 AI - General Problem Solving

Problem solving has been the key area of concern for Artificial Intelligence.

Problem solving is a process of generating solutions from observed or given data. It is however not always possible to use direct methods (i.e. go directly from data to solution). Instead, problem solving often needs to use indirect or modelbased methods.

General Problem Solver (GPS) was a computer program created in 1957 by Simon and Newell to build a universal problem solver machine. *GPS* was based on Simon and Newell's theoretical work on logic machines. *GPS* in principle can solve any formalized symbolic problem, such as theorems proof and geometric problems and chess playing. *GPS* solved many simple problems, such as the Towers of Hanoi, that could be sufficiently formalized, but ***GPS could not solve any real-world problems***.

To build a system to solve a particular problem, we need to:

- Define the problem precisely – find input situations as well as final situations for an acceptable solution to the problem
- Analyze the problem – find few important features that may have impact on the appropriateness of various possible techniques for solving the problem
- Isolate and represent task knowledge necessary to solve the problem
- Choose the best problem-solving technique(s) and apply to the particular problem

Problem definitions

A problem is defined by its '*elements*' and their '*relations*'. To provide a formal description of a problem, we need to do the following:

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

These states are called *goal states*.

Specify a set of *rules* that describe the actions (*operators*) available.

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

- *Search* is fundamental to the problem-solving process.
- *Search* is a general mechanism that can be used when a more direct method is not known.
- *Search* provides the framework into which more direct methods for solving subparts of a problem can be embedded. A very large number of AI problems are formulated as search problems.
- Problem space

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

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

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

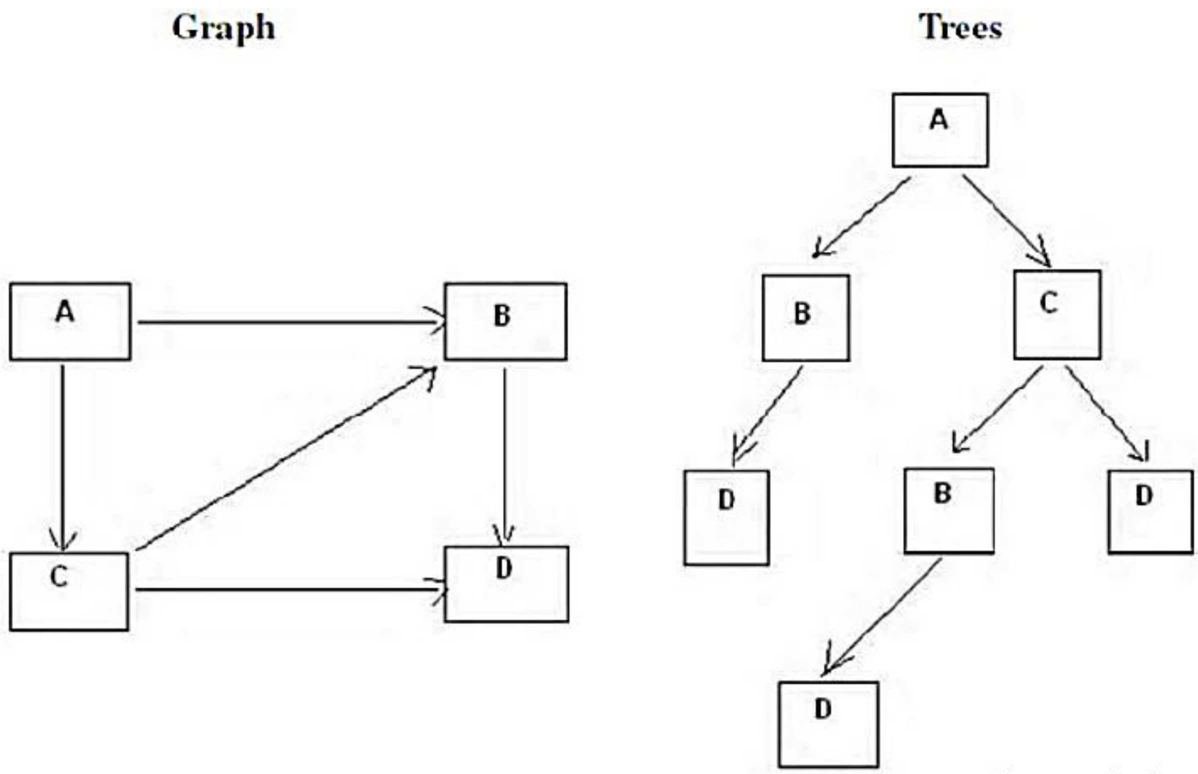


Fig. 2.1 Graph and Tree

• **Problem solving:** The term, Problem Solving relates to analysis in AI. Problem solving may be characterized as a systematic search through a range of possible actions to reach some predefined goal or solution. Problem-solving methods are categorized as *special purpose* and *general purpose*.

• A *special-purpose method* is tailor-made for a particular problem, often exploits very specific features of the situation in which the problem is embedded.

• A *general-purpose method* is applicable to a wide variety of problems. One General-purpose technique used in AI is *'means-end analysis':* a step-bystep, or incremental, reduction of the difference between current state and final goal.

2.3 DEFINING PROBLEM AS A STATE SPACE SEARCH

To solve the problem of playing a game, we require the rules of the game and targets for winning as well as representing positions in the game. The opening position can be defined as the initial state and a winning position as a goal state. Moves from initial state to other states leading to the goal state follow legally. However, the rules are far too abundant in most games— especially in chess, where they exceed the number of particles in the universe. Thus, the rules cannot be supplied accurately and computer programs cannot handle easily. The storage also presents another problem but searching can be achieved by hashing.

The number of rules that are used must be minimized and the set can be created by expressing each rule in a form as possible. The representation of games leads to a state space representation and it is common for well-organized games with some structure. This representation allows for the formal definition of a problem that needs the movement from a set of initial positions to one of a set of target positions. It means that the solution involves using known techniques and a systematic search. This is quite a common method in Artificial Intelligence.

2.3.1 State Space Search

A *state space* represents a problem in terms of *states* and *operators* that change states.
A state space consists of:

- A representation of the *states* the system can be in. For example, in a board game, the board represents the current state of the game.
- A set of *operators* that can change one state into another state. In a board game, the operators are the legal moves from any given state. Often the operators are represented as programs that change a state representation to represent the new state.
- An *initial state*.
- A set of *final states*; some of these may be desirable, others undesirable. This set is often represented implicitly by a program that detects terminal states.

• **Structure of a state space**

The *structures* of a *state space* are *trees* and *graphs*.

- ✓ A *tree* is a hierarchical structure in a graphical form.
- ✓ A *graph* is a non-hierarchical structure.

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

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

• The *Operators* are directed *arcs* between nodes.

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

• **Problem solution**

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

- ✓ A *solution cost function* assigns a numeric cost to each *path*; it also gives the cost of applying the *operators* to the *states*.
- ✓ A *solution quality* is measured by the *path cost function*; and an optimal solution has the lowest path cost among all solutions.
- ✓ The *solutions* can be *any or optimal or all*.
- ✓ The importance of cost depends on the problem and the type of solution asked

2.4 PRODUCTION SYSTEMS

Production systems provide appropriate structures for performing and describing search processes. A production system has four basic components as enumerated below.

- A set of rules each consisting of a left side that determines the applicability of the rule and a right side that describes the operation to be performed if the rule is applied.
- A database of current facts established during the process of inference.

- A control strategy that specifies the order in which the rules will be compared with facts in the database and also specifies how to resolve conflicts in selection of several rules or selection of more facts.
- A rule firing module.

The production rules operate on the knowledge database. Each rule has a precondition—that is, either satisfied or not by the knowledge database. If the precondition is satisfied, the rule can be applied. Application of the rule changes the knowledge database. The control system chooses which applicable rule should be applied and ceases computation when a termination condition on the knowledge database is satisfied.

PROBLEM CHARACTERISTICS

Heuristics cannot be generalized, as they are domain specific. Production systems provide ideal techniques for representing such heuristics in the form of IF-THEN rules. Most problems requiring simulation of intelligence use heuristic search extensively. Some heuristics are used to define the control structure that guides the search process, as seen in the example described above. But heuristics can also be encoded in the rules to represent the domain knowledge. Since most AI problems make use of knowledge and guided search through the knowledge, AI can be described as *the study of techniques for solving exponentially hard problems in polynomial time by exploiting knowledge about problem domain.*

To use the heuristic search for problem solving, we suggest analysis of the problem for the following considerations:

- Decomposability of the problem into a set of independent smaller subproblems
- Possibility of undoing solution steps, if they are found to be unwise
- Predictability of the problem universe
- Possibility of obtaining an obvious solution to a problem without comparison of all other possible solutions
- Type of the solution: whether it is a state or a path to the goal state
- Role of knowledge in problem solving
- Nature of solution process: with or without interacting with the user

CHARACTERISTICS OF PRODUCTION SYSTEMS

Production systems provide us with good ways of describing the operations that can be performed in a search for a solution to a problem.

At this time, two questions may arise:

1. Can production systems be described by a set of characteristics? And how can they be easily implemented?
2. What relationships are there between the problem types and the types of production systems well suited for solving the problems?

To answer these questions, first consider the following definitions of classes of production systems:

1. A monotonic production system is a production system in which the application of a rule never prevents the later application of another rule that could also have been applied at the time the first rule was selected.
2. A non-monotonic production system is one in which this is not true.
3. A partially communicative production system is a production system with the property that if the application of a particular sequence of rules transforms state P into state Q, then any combination of those rules that is allowable also transforms state P into state Q.
4. A commutative production system is a production system that is both monotonic and partially commutative.

Table 2.1 Four Categories of Production Systems

Production System	Monotonic	Non-monotonic
Partially Commutative	Theorem Proving	Robot Navigation
Non-partially Commutative	Chemical Synthesis	Bridge

Is there any relationship between classes of production systems and classes of problems? For any solvable problems, there exist an infinite number of production systems that show how to find solutions. Any problem that can be solved by any production system can be solved by a commutative one, but the commutative one is practically useless. It may use individual states to represent entire sequences of applications of rules of a simpler, non-commutative system. In the formal sense, there is no relationship between kinds of problems and kinds of production systems Since all problems can be solved by all kinds of systems. But in the practical sense, there is definitely such a relationship between the kinds of problems and the kinds of systems that lend themselves to describing those problems.

Partially commutative, monotonic productions systems are useful for solving ignorable problems. These are important from an implementation point of view without the ability to backtrack to previous states when it is discovered that an incorrect path has been followed. Both types of partially commutative production systems are significant from an implementation point; they tend to lead to many duplications of individual states during the search process. Production systems that are not partially commutative are useful for many problems in which permanent changes occur.

Types of Production System:

An AI production system has three main elements which are as follows:

- **Global Database:** The primary database which contains all the information necessary to successfully complete a task. It is further broken down into two parts: temporary and permanent. The temporary part contains information relevant to the current situation only whereas the permanent part contains information about the fixed actions.
- **A set of Production Rules:** A set of rules that operates on the global database. Each rule consists of a precondition and post condition that the global database either meets or not. For example, if a condition is met by the global database, then the production rule is applied successfully.
- **Control System:** A control system that acts as the decision-maker, decides which production rule should be applied. The Control system stops computation or processing when a termination condition is met on the database.

Features of a Production System:

A production system has the following features:

1. **Simplicity:** Due to the use of the IF-THEN structure, each sentence is unique in the production system. This uniqueness makes the knowledge representation simple to enhance the readability of the production rules.
2. **Modularity:** The knowledge available is coded in discrete pieces by the production system, which makes it easy to add, modify, or delete the information without any side effects.
3. **Modifiability:** This feature allows for the modification of the production rules. The rules are first defined in the skeletal form and then modified to suit an application.
4. **Knowledge-intensive:** As the name suggests, the system only stores knowledge. All the rules are written in the English language. This type of representation solves the semantics problem.