جامعة شط العرب

University Of Shatt Al-Arab

College of Computer Science

Artificial Intelligence

Third Stage

2024

Lecture 1: Introduction Artificial Intelligence

What is intelligence?

Intelligence can be defined as the ability to acquire, process, and apply knowledge, particularly knowledge derived from the natural sciences, in addition to the ability to think logically and solve problems based on this information. It is clear that designing an intelligent program goes beyond merely developing algorithms; it requires simulating the cognitive processes humans rely on to understand and interact with the world.

What is Artificial Intelligence?

With the rapid advancement of technology, artificial intelligence (AI) has become a common term in everyday life. From chatbots and virtual assistants to self-driving cars and recommendation algorithms, AI's impact is visible in numerous aspects of modern living. Artificial Intelligence (AI) refers to the development of computer systems of performing tasks that require human intelligence. AI aids, in processing amounts of data identifying patterns and making decisions based on the collected information. This can be achieved through techniques like Machine Learning, Natural Language Processing, Computer Vision and Robotics. AI encompasses a range of abilities including learning, reasoning, perception, problem solving, data analysis and language comprehension. The ultimate goal of AI is to create machines that can emulate capabilities and carry out diverse tasks, with enhanced efficiency and precision. The field of AI holds potential to revolutionize aspects of our daily lives.

Definition of artificial intelligence

Intelligence has been defined in many different ways.

- Artificial intelligence, as a branch of computer science, focuses on studying and teaching machines intelligent human behaviors. This involves programming computers to think, solve problems, and make decisions in a way that mimics human abilities. Thanks to this programming, computers can recognize images and shapes, understand and generate natural language, play chess, diagnose diseases, and perform other tasks that require intelligence.

- It is the science that explores how to make computers perform tasks typically done by humans, but in a more efficient and effective manner (Elaine Rich).

- It is the science focused on creating machines that can carry out tasks requiring a level of human intelligence when done by people (Marvin Minsky).

- It is the ability of machines to perform tasks that require human intelligence when performed, such as logical reasoning and learning(Martin Weik).

Comparison between human intelligence and artificial intelligence

Here's a comparison table between **Human Intelligence** and **Artificial Intelligence (AI)** based on several factors:

Aspect	Human Intelligence	Artificial Intelligence
Learning	Learns through experience, education, and social interactions.	Learns through data input, training algorithms, and patterns.
Creativity	Can generate novel ideas, think abstractly, and innovate.	Limited creativity; can generate solutions based on existing data but lacks true innovation.
Emotions	Possesses emotions, empathy, and subjective experiences.	Does not experience emotions or subjective states.
Problem-Solving	Approaches problems with creativity, intuition, and logic.	Solves problems based on programmed algorithms and heuristics.
Speed of Processing	Relatively slower, depends on biological processing.	Extremely fast, limited by computational power and algorithms.
Decision-Making	Considers emotions, ethics, intuition, and experience.	Decisions are made based on data and algorithms, without ethical or emotional considerations.
Error Handling	Can learn from mistakes and adapt behavior.	Can only correct errors have based on predefined training or programming.

This table highlights the unique strengths and limitations of both human intelligence and artificial intelligence across different domains. While AI excels in speed, processing power, and data handling, human intelligence remains unparalleled in creativity, emotion, adaptability, and ethical reasoning.

AI problems

The problem of simulating intelligence is broken down into a number of specific sub-problems. These problems consist of specific features or capabilities that researchers would like to embody in an intelligent system. The features listed below have received the most attention:

1. **Knowledge representation**

   It is the focus of artificial intelligent research. Many of the problems that are expected to be solved by machines they will require extensive knowledge of the world. Among the things that need to be represented by Artificial Intelligent: objects and properties and groups taxonomic relationships between objects; and attitudes and events, states and time; causes and effects; knowledge of knowledge (what we know about what people know) and many other areas.

2. **Widening logical knowledge**

   The average person knows a large number of facts about corn. Research projects that seek to build a complete knowledge base of logical, it must be built in a traditional manner where it is building the complex concepts, one by one. One of the main objectives that the computer understands a multitude of concepts to be able to learn to read through sources such as the Internet.

3. **Planning**

You must be smart agents will be able to set goals and achieve them they needed a way to visualize the future (you must have the ability to represent the case of human beings in this world, and be able to predict the extent of their ability to change it), and be able to choose to maximize the benefit.

4. **Learning**

The machine learning pivotal in artificial intelligent researches since the beginning. Unsupervised learning is the ability to find patterns in a large number of inputs. Supervised learning includes both classification (ability to determine to which category something belongs in, after seeing a number of models for a number of things from several categories), and retreat (the discovery of the mechanism of continuing that will generate the outputs of the input, in light of the range of input and output numerical examples).

5. **Mechanism of natural language**

Natural language processing gives machines the ability to read and understand the languages spoken by humans. Many researchers hope that the natural language processing system that is strong enough to gain knowledge of his own, by reading the existing text available over the Internet. Some direct applications of natural language processing include information retrieval (or text) analysis.

6. **Movement and the possibility of change**

The field of robotics is closely related to artificial intelligent. Necessary robots intelligent to be able to handle tasks such as object manipulation and navigation, in light of localization sub-problems (that you know where you are), mapping (to know what is around you) and plan the movement (to know how to get there).

7. **Perception**

Machine perception is the ability to use input from sensors (such as cameras, microphones, sonar and other more exotic) to draw out aspects of the world. Computer vision is the ability to input visual analysis. Sub few of the problems: speech recognition face recognition and identification of objects.

Lecture 2: AI characteristics, goals, applications

Characteristics of AI

There are many characteristics:

1- Representability of knowledge

This means using special structures to represent knowledge represented in the form of facts, relationships, laws, frameworks, ... and other forms of knowledge. The set of knowledge structures used in the description constitutes what is called the knowledge base and this knowledge may be:

- Complete description of all the details of the problem to be solved.
- Incomplete or inaccurate description
- Containing a contradictory or conflicting description

Artificial intelligence software is concerned with finding solutions according to the nature of the represented issue. The process of representing knowledge must precede the process of obtaining knowledge, in which the focus is on the method of obtaining or collecting knowledge by emphasizing a specific area of the issue.

2- Symbolic processing

It represents the ability of artificial intelligence software to deal with non-numerical symbols in addition to numerical values. This feature enables this type of software to deal with knowledge in its natural form through its ability to process descriptively as well as accurately digitally.

3- Inference Making

It is the ability of artificial intelligence software to deduce possible solutions, especially for problems that cannot be solved using traditional algorithms. Such capabilities are programmed by storing the available knowledge about all possible and probable solutions.

4-Meta Knowledge

It is the knowledge required to manage thinking strategies effectively and without it, the efficiency of thinking is reduced and in complex situations, thinking may become impossible. This knowledge divides the problem into smaller sub-problems or divides the main goals of the system into sub-goals. By solving these partial problems or sub-goals, the overall problem is solved or the main goals are achieved.

5- Declarative Language

These are languages that facilitate the process of representing knowledge and also help in the process of reasoning, and for this purpose these languages are provided with natural knowledge representation capabilities, such as representing facts, conditional procedures, complex relationships, and other capabilities.

6- Learning Mechanisms

Learning techniques provide the computer with methods that enable an intelligent system to learn or develop cognitively, either by recognizing patterns through repeated behaviors, or by:

a) Analyzing differences.

b) Learning from examples.

c) Training neural networks.

d) Simulating evolution to learn specific evolutionary paths, using algorithms like genetic algorithms.

7- Goals Searching

It includes procedures that provide the system with the ability to plan and achieve specific goals by setting criteria for generating planning strategies. These strategies depend on producing all possible results at each step of the solution and forming a searchable structure.

8- Interactive capabilities that simulate human capabilities

These capabilities enable interaction between intelligent systems and users, allowing communication through voice responses, robotic arms, or a series of questions and answers. They can also include recognizing and responding to images provided by the user, or diagnosing and analyzing content. This type of interaction, which mirrors how humans interact with each other, is referred to as human-mimicking interaction.

Goals Of AI

The primary goals of Artificial Intelligence (AI) include:

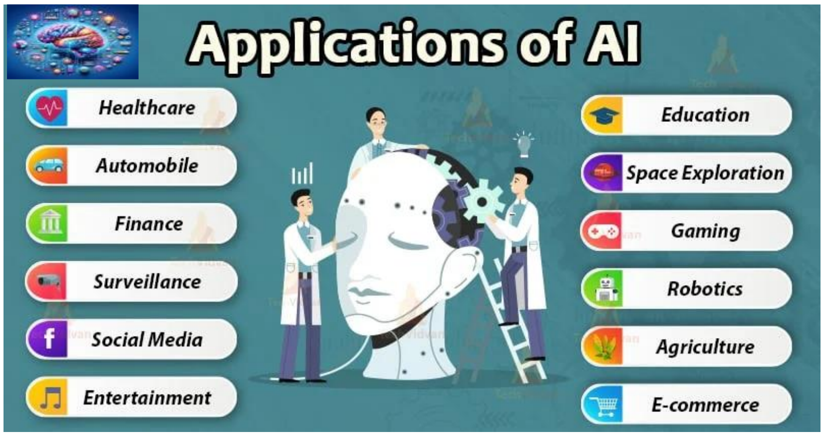
1. **Automation of Tasks**: Developing systems that can perform tasks traditionally requiring human intelligence, such as decision-making, problem-solving, and pattern recognition.
2. **Learning and Adaptation**: Enabling machines to learn from data and experiences, allowing them to improve performance over time without human intervention.
3. **Perception and Interaction**: Enhancing the ability of machines to perceive and interpret sensory inputs (e.g., vision, speech), and to interact naturally with humans and the environment.
4. **Reasoning and Problem Solving**: Designing systems that can reason logically, make decisions based on available information, and solve complex problems effectively.
5. **Creativity and Innovation**: Enabling AI to generate new ideas, concepts, and solutions in areas such as art, design, and technology.
6. **Ethics and Safety**: Ensuring that AI operates within ethical boundaries, maintains user safety, and makes decisions that align with societal values.

Applications of AI

Artificial intelligent has been used in a wide range of fields including:

1. Healthcare

AI improves diagnosis, treatment, patient care, and hospital efficiency by analyzing medical images, personalizing treatment plans, and assisting in robotic surgeries. It accelerates drug development and optimizes hospital administration.



2. Automobile

AI advances self-driving cars, enhances safety with features like automatic braking and lane-keeping, and optimizes logistics and manufacturing processes for greater efficiency.

3. Finance

AI boosts fraud detection, risk management, and customer service by analyzing transactions, executing trades, and handling credit scoring and customer inquiries.

4. Surveillance

AI enhances security through facial recognition, behavior analysis, and real-time monitoring, assisting in identifying threats and predicting criminal activities.

5. Social Media

AI curates personalized content, detects harmful behavior, and enhances engagement by analyzing user data, moderating content, and optimizing ads and influencer marketing.

6. Entertainment

AI personalizes recommendations, creates content, and improves production quality in movies and games. It also drives interactive experiences in VR and AR.

7. Education

AI personalizes learning with adaptive platforms, provides instant feedback through virtual tutors, and automates grading. It supports curriculum development by analyzing trends and enhances language learning with interactive chatbots.

8. Space Exploration

AI aids in navigation and autonomous decision-making for spacecraft and rovers, analyzes data from space missions, and assists in discovering celestial objects and predicting space weather.

9. Gaming

AI enhances gameplay with adaptive NPCs, generates unique game content, improves graphics and physics, and offers strategic insights in competitive gaming.

10. Robotics

AI enables robots to perform complex tasks autonomously in manufacturing, healthcare, and other sectors. It assists with navigation, precision tasks, and operates in environments like warehouses or disaster zones.

11. Agriculture

AI optimizes crop management by analyzing data from sensors and satellites, helps with precision farming, and uses drones and robots for monitoring, planting, and harvesting.

12. E-commerce

AI improves customer experience with personalized recommendations, chatbots for support, inventory management, and logistics. It also adjusts pricing based on demand and competition.

Lecture 3: Knowledge representation (the concept of knowledge base and methods of its representation)

What is knowledge?

It is information about a particular field or subject, or it is information about how to do something, or it can be defined as information in a moral context.

The identifier can take several forms, for example:

Eva is a female.

Adam is male.

Female with children is a mother.

Fathers are male.

Knowledge Representation

Knowledge representation is a branch of artificial intelligence that refers to the way information and knowledge are formulated and stored within a computer system so that the system can use it to understand the world and make decisions. The goal of knowledge representation is to transform human knowledge into a form that a computer can process. To create programs that have "intelligent" qualities, it is necessary to develop techniques for representing knowledge. Unlike to people, computers do not have the ability to acquire knowledge on their own.Al programs use structures called knowledge structures to represent objects, facts, rules, relationships, and procedures. The main function of the knowledge structure is to provide the needed expertise and information so that a program can operate in an intelligent manner. Knowledge structures are usually composed of both traditional data structures and other complex structures such as Logical, frames, scripts, semantic networks, conceptual graph, and ATN (augment transition network).

Any method of representing knowledge has the following characteristics:

1- **Representation Adequacy: -**

It should be able to represent all necessary identifiers relevant to the given problem domain.

2- **Inferential Ability: -**

The ability to process represented knowledge in a way that allows the production of new knowledge.

3- **Inferential Efficiency: -**

The ability to include additional information into the knowledge structure that can be used to guide the inference mechanism towards a better path.

4- **Acquisitional Efficiency: -**

The ability to acquire new information easily, in other words (automatically update the structure with new knowledge).

Conditions for representing knowledge in artificial intelligence: -

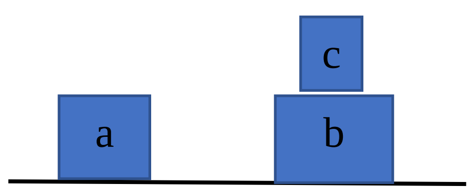
1) Descriptive Identifier Processing Example: To describe the figure below using Prolog language as follows

On (a, table)

On (b, table)

On (c, b)

Left (a, b)



Table

2) Allows new knowledge to be derived from a basic set of facts.

Ali is Ahmed father

Father is a male

You can conclude

Ali is male

3) Allowing the representation of general principles as well as the ability to represent special cases.
   Is_a (bird, animal).
   Is_a (canary, bird).
   Is_a (Toki, canary)

4) Allows extraction of complex semantic meaning.
   The sentence below requires knowledge of the semantic meaning of each word in the sentence to discover the correctness or incorrectness of the linguistic sentence.

   (The cat is studying the lesson)

5) Allowing meta-knowledge representation.

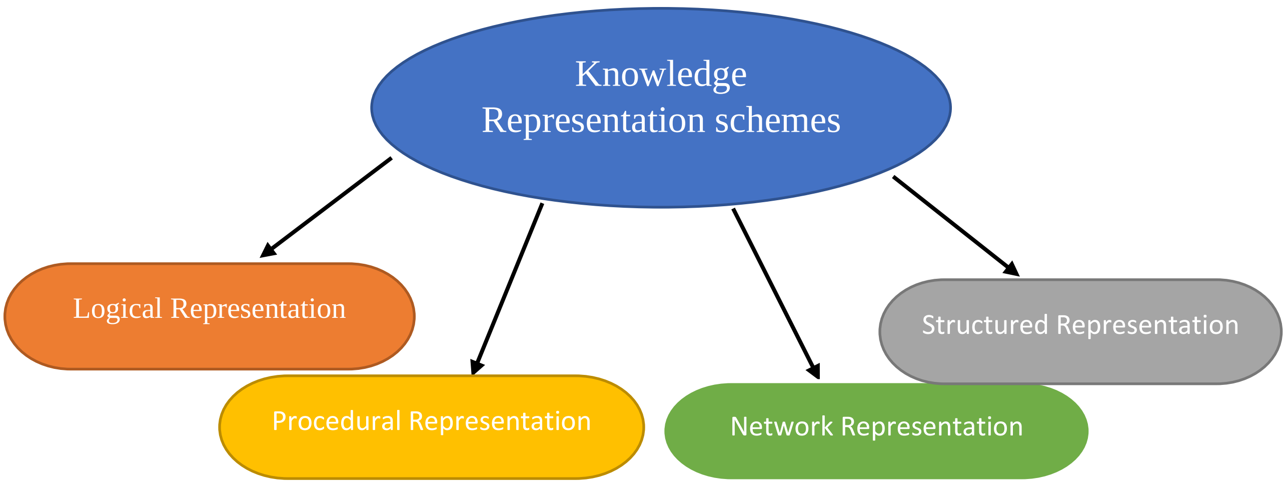
Knowledge Classification

It is useful to study the types of knowledge that a knowledge system can represent, as classified by artificial intelligence scientists as follows:

a. **Object:** Any intelligent system must have the ability to represent entities and properties of the physical world.
b. **Events:** This category includes the representation of events, actions, and behavior that exist in the real world.
c. **Performance:** This category includes information on how to accomplish a particular task, such as how to construct a sentence.
d. **Meta- Knowledge:** It is the knowledge that explains the represented knowledge

Knowledge representation schemes

In Al, there are four basic categories of representational schemes: logical, procedural, network and structured representation schemes.



1. Logical representation

uses expressions in formal logic to represent its knowledge base. **Predicate Calculus** is the most widely used representation scheme.

2. Procedural representation

represents knowledge as a set of instructions for solving a problem. These are usually **if-then rules** we use in rule-based systems.

3. Network representation

captures knowledge as a graph in which the nodes represent objects or concepts in the problem domain and the arcs represent relations or associations between them. **Semantic networks and conceptual graph** are example of this scheme

4. Structured representation

extends network representation schemes by allowing each node to have complex data structures named slots with attached values, these values may be simple numeric or symbolic data, pointer to other frames or even procedures, **scripts and frames** are examples of this scheme. We will focus on logic representation schemes in this chapter.

Lecture 4: Knowledge Representation (Study of Types of Knowledge Representation in AI)

Knowledge Representation

Knowledge representation methods are divided into the following categories:

1- Logical representation

Logic: is the study of principles and rules that govern reasoning and argumentation. It provides a systematic way to evaluate the validity of statements and arguments, determining whether conclusions follow logically from premises.

There are several types of logic, including:

a) Propositional Logic

Propositional logic is a subset of predicate logic, often applied to prove theorems by using inference laws to transform a set of statements into more complex ones.

This type of representation can be used some of conceptions such as:

- Axiom: it always known as truth.
- Proposition: it’s known as a Boolean sentence, the truth symbols maybe true or false,
- Theorem: it’s a Boolean sentence; it can conclude form the axioms.

propositional logic focuses exclusively on declarative sentences, where the answers are either (yes or no). Symbols in propositional logic are typically represented as (p, q, r).

The propositional logic is content form the following three parts:

- A set of the concepts, axioms and proposition that can be represented by **Well-Formed Formula (WFF)**. It symbols denote *propositions of* statements about the world that may be either true or raise, such as "the car is red" or "water is wet." WFFs are denoted by uppercase letters near the end of the English alphabet (i.e., P, Q …etc.).

- A set of connections that can connect two or more WFF sentences:

NOT	˥	Negation Connection
AND	∧	Conjunction Connection
OR	∨	Disjunction Connection
IF	→	Implication Connection A IF B was write as A→B
IFF	↔	Equivalence Connection A IFF B was write as A↔B

- A set of Inference Rules that can be used to conclude new sentences form the old sentences. We can explain two types of these inference rules:

- **Modus Ponens (MP)**

  If the sentences P and (P→Q) are known to be True, then this rule infers Q is True.

  Structure of Modus Ponens:

- **Modus Tolen (MT)**

  If the sentence (P→Q) is known to be True and Q is known to be False, Then this rule infers ¬P is False.

b) Predicate Logic

Predicative logic, or first-order logic, or relational calculus, is a formal language in which propositions are expressed in terms of predicates, variables, and determinants. It differs from propositional logic, which lacks determinants.

Key Components of Predicate Logic:

- **Predicates:** Functions or properties that can be applied to objects. A predicate takes an object (or objects) as input and returns true or false.
  Example: P(x) could mean "x is a student."
- **Variables**: Symbols that represent objects in the domain of discourse.
  Example: x, y, z
- **Quantifiers**: Symbols that express the scope of the variables.
  - **Universal Quantifier** (∀): Means "for all.
  - **Existential Quantifier** (∃): Means "there exists."

Lecture 5: Theorem Proving in Artificial Intelligence

Introduction

Theorem proving in AI is a field of study that focuses on developing algorithms and systems that can automatically prove mathematical theorems or logically verify statements. It combines elements of formal logic, computer science, and mathematical logic to create tools that can assist or automate the entire process of proving or disproving propositions. Theorem proving lies at the heart of automated reasoning, where AI systems emulate logical reasoning. Theorem proving in AI is a powerful tool that enables the formal verification of systems, the automation of reasoning tasks, and even the discovery of new mathematical insights. By automating logical reasoning, AI-powered theorem-proving tools offer solutions to problems in critical areas such as software and hardware verification, mathematics, and logic. Despite challenges related to complexity and scalability, advancements in this field continue to push the boundaries of what these systems can achieve, moving AI closer to fully replicating human reasoning.

What is Theorem Proving?

Theorem proving involves the formal verification of statements (theorems) within a specified logical system. In formal logic, a theorem is a statement derived from a set of axioms through the application of logical rules. The goal of theorem proving is to define a logical sequence that proves or disproves a theorem by demonstrating the existence or non-existence of such a sequence. In the context of artificial intelligence, theorems are automatically proven, allowing for wide applications in fields where validity and formal verification are critical, such as software engineering, hardware design, and mathematics. Simply put, theorem proving is the ability of computers to automatically prove theorems using formal logic. It is a branch of automated reasoning, a subfield of artificial intelligence whose primary task is to automatically prove or disprove mathematical statements and propositions.

Theorem proving is a major area of research in automated reasoning, providing the ability to automatically verify mathematical theorems. This process is invaluable in various fields, as it enhances the speed, efficiency, and accuracy of theorem verification compared to manual methods. With recent advances in machine learning, cognitive computing, and artificial intelligence, theorem proving has gained significant momentum. It plays a crucial role in fields such as computer science, mathematics, physics, engineering, and artificial intelligence. This article aims to introduce the basics of theorem proving for those interested in exploring this exciting field of study.

Why Theorem Proving is Important in AI

Automating Reasoning: Theorem proving represents one of the core aspirations of AI to replicate human reasoning in a formal, logical manner. By automating the process of proving theorems, AI systems can perform tasks like verifying the correctness of algorithms, proving properties of programs, and solving mathematical problems.

Formal Verification: Theorem proving is widely used in formal verification, which ensures that a system (software or hardware) adheres to its specification. This is crucial in industries where system failures could have catastrophic consequences, such as aerospace, medical devices, and cryptography.

Mathematical Discoveries: Theorem proving systems also play a role in modern mathematical research. They assist mathematicians in proving highly complex theorems or verifying proofs that are too lengthy or intricate to be checked manually.

Challenges in Theorem Proving

1. **Computational Complexity**: The space of possible proofs is often vast, especially in first-order or higher-order logic. Theorem proving is NP-complete in propositional logic and undecidable in first-order logic, meaning there is no general algorithm guaranteed to find a proof in finite time.

2. **Decidability**: While some logical systems are decidable (where a proof or disproof can always be found), others, like first-order logic, are undecidable. In these cases, a theorem prover may search indefinitely without ever finding a solution.
3. **Scalability**: Automated theorem provers struggle to scale efficiently to large, complex systems. In interactive theorem proving, human input helps, but even then, formalizing large theorems or systems can be a time-consuming and error-prone task.

Basic Concepts in Theorem Proving

1-Formal Logic: The foundation of theorem proving is formal logic, which provides the rules and syntax for creating statements and deriving conclusions. Common logical systems used in theorem proving include:

- **Propositional Logic**: Deals with simple statements that are either true or false.
- **First-Order Logic (FOL)**: More expressive than propositional logic, FOL includes quantifiers (like ∀ and ∃) and predicates to describe properties and relationships among objects.

2-Axioms and Inference Rules: A theorem is proven by starting from a set of axioms (assumed truths) and applying inference rules to derive new truths. Common inference rules include:

- **Modus Ponens**: If $A \rightarrow B$ and A are true, then B is also true.

Advantages of Proving Theorems

Proving theorems is a powerful technique that has many advantages over traditional proof methods. These advantages include:

- **Automated process:** Proving theorems is an automated process that is faster and more efficient than manual proof methods.

- **Improvement**: Proving theorems can help improve complex systems that are prone to errors. By automatically identifying and fixing errors, proving theorems helps improve the effectiveness and efficiency of these systems.
- **Correctness:** Proving theorems can prove the correctness of a system or algorithm and uncover errors that may be difficult to identify through manual methods.

Types of Theorems Proving in Artificial Intelligence

There are two primary types of theorems proving approaches in Artificial Intelligence (AI)**:** **Automated Theorem Proving (ATP)** and **Interactive Theorem Proving (ITP)**. Each of these approaches plays a crucial role in various applications where formal logic and reasoning are needed, such as in software verification, hardware design, and mathematical proofs.

Automated Theorem Proving (ATP): -

It is also called the solution technique for proving the theorem in the calculus of propositions and predicates, which attempts to show that the negation of a statement produces a contradiction with known statements. This technique relies on the refutation that will occur in the knowledge base (KB).

The proofs of the algorithmic solution technique involve the following steps:

1. Assume that ┐P is true.

2. Show that the basic axioms with ┐P lead to a contradiction.

3. Conclude that since the axioms are true, ┐P must be false.

4. Since ┐P is false, P must be true.

Before done this algorithm, it must be converting all the sentences form WFF to Clause form. Therefore, it can use the following algorithm to done this convert:

Algorithm to convert a WFF to Clause Form:

1.Change P↔Q to ┐P∨Q

P↔Q to (┐P∨Q) ∧ (┐Q∨P)

2. Reduce the range of negative; for example, covert

$\neg(\neg a) \equiv a$

$\neg(\exists X)a(X) \equiv (\forall X)\neg a(X)$

$\neg(\forall \mathrm{X})\mathrm{b}(\mathrm{X}) \equiv (\exists \mathrm{X})\neg b(X)$

$\neg(a \land b) \equiv \neg a \lor \neg b$

3.Relocate the universal quantifier ∀ to front of the clauses. Example:

$\forall X\ \forall Y\ (P(X) \lor \neg Q(Y))$ to $\forall X\ P(X) \lor \forall Y\ \neg Q(Y)$

4. Rewrite the sentence in conjunction normal form (i.e., the AND would be distributed with respect to the OR). For example: (A ∧ B) ∨ (B ∧ C) would be redistribute (A∨ B) ∧ (B∨ C) ∧ B ∧ (A∨ C).

Lecture 6: Apply the theorem to a set of examples.

Ex1: You have the following axioms:

1. feather (rook).

2. $\forall X$ (feather(X)→ bird(X))

By Automatic Theorem Proving prove that bird (rook)

Firstly, change WFF to Clause form

1. feather (rook).

2. ┐ feather(X) ∨ bird(X))

Add the negative theorem that need to prove

3. ¬ bird (rook)

(1), (2) 4. bird (rook) according to MP

(3), (4) empty

So, the theorem ¬ bird (rook) must be False, therefore the theorem bird(rook) must be True.

Ex2: You have the following axioms:

1. father (ali, ahmed).

2. has (ali, money)

3. (father (Z, X) ∧ has (Z, Y)) → has (X, Y)

By Automatic Theorem Proving prove that has (ahmed, money)

Firstly, change WFF to Clause form

1. feather (ali, ahmed).

2. has (ali, money).

3. ┐ father (Z, X) ∨ ┐ has (Z, Y) ∨ has (X, Y)

Add the negative theorem that need to prove

4. ¬ has (ahmed, money)

(2), (3) 5. ┐ father (ali, X) ∨ has (X, money)

(4), (5) 6. ┐ father (ali, ahmed)

(1), (6) empty

So, the theorem ¬ has (ahmed, money) must be False, therefore the theorem has (ahmed, money) must be True.

Ex3: You have the following axioms:

1. Fido is a dog.

2. All dogs are animals.

3. All animals will die.

By Automatic Theorem Proving prove that Fido will die

Firstly, change all sentences to WFF

1. dog (fido).

2. (dog(X)→ animal(X))

3. (animal(Y)→ die(Y)).

Add the negative theorem that need to prove

4. ¬ die (fido)

After that, change WFF to Clause form

1. dog (fido).

2. ┐ dog(X) ∨ animal(X))

3. ┐ animal(Y) ∨ die(Y)).

4. ¬ die (fido)

(1),(2) 5. animal (fido) according to MP

(3),(5) 6. die(fido) according to MP

(4),(6) empty

So, the theorem ¬ die (fido) must be False, therefore the theorem Fido will die must be True.

Ex4: You have the following axioms by WFF:

1. gt (2017, 79).

2. man (ali)

3. man (X2) → dead (X2, 79)

4. Now= 2017

5. alive (X4, T3) → ┐ dead (X4, T3)

6. [dead (X4, T3) ∧ gt (T4, T3)] → dead (X4, T4)

By Automatic Theorem Proving prove that : ┐ alive (ali, Now).

Firstly, change WFF to Clause form

1. gt (2017, 79).

2. man (ali)

3. ┐ man (X2) ∨ dead (X2, 79)

4. Now= 2017

5. ┐ alive (X4, T3) ∨ ┐ dead (X4, T3)

6. ┐ dead (X4, T3) ∨ ┐ gt (T4, T3) ∨ dead (X4, T4)

Add the negative theorem that need to prove

7. alive (ali, Now)

(4),(7) 8. alive (ali, 2017)

(2),(3) 9. dead (ali, 79) according to MP

(6),(9) 10. ┐ gt (T4, 79) ∨ dead (ali, T4) according to MP

(1),(10) 11. dead (ali, 2017) according to MP

(5),(11) 12. ┐ alive (ali, 2017) according to MT

(8),(12) empty

So, the theorem alive (ali, 2017) must be False, therefore the theorem ┐ alive (ali, 2017) must be True.

Lecture 7: Mathematical induction and deduction methods

Inference Rules

Inference rules are fundamental logical principles used to derive conclusions from premises. They are key to reasoning in formal systems like mathematics, logic, and computer science.

1-Deductive: It is a way of thinking from the general to the specific, meaning that it is a way of thinking in which conclusions are built based on logical premises.

2-Inductive: It is a way of thinking from the specific to the general, meaning that it is a way of thinking that depends on a specific category to reach inference. The nature of the category.

Here are some of the most common inference rules:

1. Modus Ponens (Law of Detachment)

- If (P→ Q) (If P, then Q) is true, and P is true, then Q must be true.
- Form:

  - $P \rightarrow Q$

  - P

  - $\therefore$ Q

- Example:

- If it rains, the ground will be wet.

- It is raining.

- Therefore, the ground is wet.

2. Modus Tollens (Denying the Consequent)

- If (P →Q) is true, and Q is false, then P must be false.
- Form:

  -P →Q

  - ┐Q

  - ∴ ┐P

- Example:

  - If it rains, the ground will be wet.

  - The ground is not wet.

  - Therefore, it is not raining.

3. Disjunction Elimination (Or Elimination)

- If P ∨ Q (either P) or Q) is true, and P is false, then Q must be true.
- Form:

  - P ∨ Q

  - ┐P

  - ∴ Q

- Example:

  - Either I will go to the party, or I will stay home.

  - I am not going to the party.

  - Therefore, I will stay home.

4. Hypothetical Syllogism (Chain Rule)

- If $P \rightarrow Q$ and $Q \rightarrow R$ are true, then $P \rightarrow R$ is true.
- Form:

  - $P \rightarrow Q$

  - $Q \rightarrow R$

  - $\therefore P \rightarrow R$

- Example:

- If I study, I will pass the exam.

- If I pass the exam, I will graduate.

- Therefore, if I study, I will graduate.

5. Disjunctive Syllogism

- If $P \lor Q$ is true, and P is false, then Q must be true.

- Form:

  - $P \lor Q$

  - $\neg P$

  - $\therefore Q$

- Example:

  - Either it will rain or it will snow.

  - It did not rain.

  - Therefore, it snowed.

Lecture 8: Blind Search, Heuristic Search

Problem Spaces

Problem Spaces (or Solution Spaces) is a concept used in mathematics, artificial intelligence, and computational sciences to describe all possible states or solutions that can exist for a given problem. In other words, a problem space is the set of all possible conditions or situations that can occur during the search for a solution to a problem.

The basic components of a problem space:

1. Initial State:

This is the starting point from which the search for the solution begins. In games, for example, the initial state is the initial setup of the game before any moves are made.

2. Possible States:

These represent all the states that the system can reach based on the transitions or steps that can be taken from the initial state.

3. Transitions:

These are the rules or operations that allow you to move from one state to another in the problem space. These operations may be a set of defined steps or transformation rules depending on the problem.

4. Goal State:

The goal state is the condition that represents the final solution to the problem. In this state, the objective is achieved, and the solution criteria are met.

5. Solution Path:

This is the sequence of transitions that takes you from the initial state to the goal state. The objective of searching through the problem space is to find this path.

Search Methods

Search in artificial intelligence systems is an important method for solving problems, and it consists of two main processes:

1. State Description:

This process involves identifying and representing all possible states that may exist in the problem, including the initial state (the start), intermediate states, and the goal. The state description is based on determining what information is needed to understand the problem's situation.

2. Search Strategy Selection:

After describing the state, an appropriate search strategy is chosen to navigate through these states in order to reach the solution. The choice of search strategy depends on the nature of the problem and the available information.

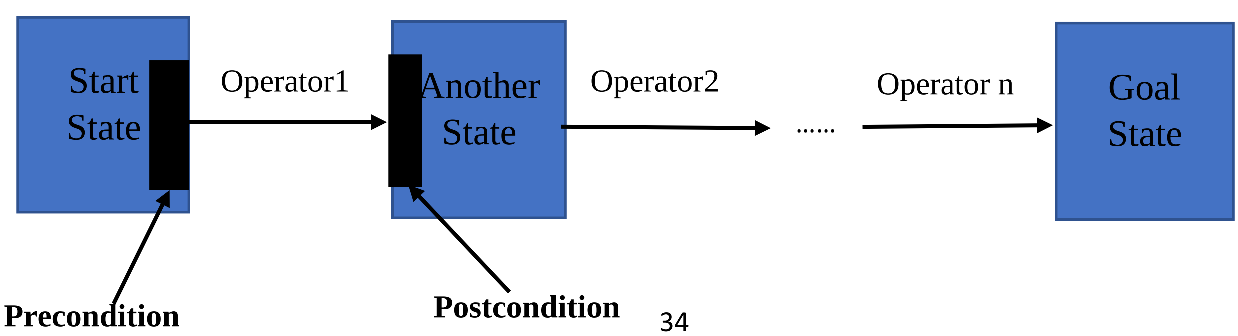
To describe a problem, we need to define the following terms:

1. State: A state is a description of a certain situation in the real world using some facts.

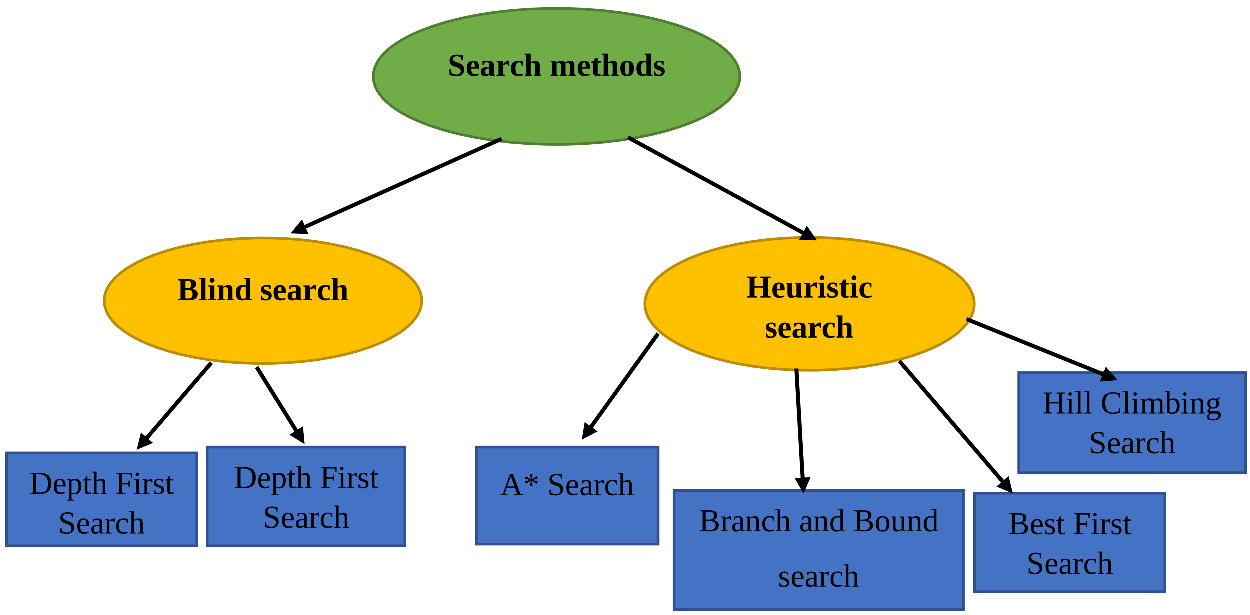
2. Operator (Action): An operator is a transition function from one state to another that brings us closer to the goal.

3. Conditions: Operators are usually accompanied by a set of conditions, which may include:

- **Preconditions:** These are conditions that must be met (i.e., true) before the operator can be applied. They are useful because they define the necessary requirements for applying the operator.
- **Postconditions:** These are conditions that must be met (i.e., true) after the operator is applied. They are useful because they summarize the effects of the operator.



Search methods are divided into two main types:



1. Blind Search:

Blind search in this search we generate a possible solution. When we find the solution, we are done as we will repeat it by trying another possible case. This search is called "generate and test" because we generate a possible solution and then test it. Without a suitable solution we try again. This type of search takes all the nodes of the tree in a specific order until it reaches the goal. There are many search strategies that fall under the title of "deep search". This term means that they do not have additional information about the cases other than that mentioned in the problem definition. All they can do is generate alternatives and distinguish between target and non-target cases. Therefore, blind search strategies do not have any prior information about the goal and simple paths do not lead to it. However, blind search is not bad as more problems or applications require a solution. In other words, there are some problems that provide good solutions if they are solved using deep or comprehensive search

There are two types of blind search:

a) Depth First Search (DFS)
b) Depth First Search (BFS)

2. Heuristic Search:

In heuristic search, one or more heuristic functions are generally used to select the best candidate cases from a set of available cases that can be generated from a known case. In other words, the heuristic function measures the suitability of the candidate nodes. The better the selection of nodes, the fewer intermediate cases are needed to reach the goal. It is a method that may not always find the best solution, but it is guaranteed to find a good solution in a reasonable time. Heuristic search is useful in solving the following problems: It cannot be solved in any other way: It takes an infinite amount of time or a very long time to compute. The heuristic search method generates and tests algorithms. Heuristic search methods use an evaluation function to guide the search toward the goal.

There are three types of heuristic search:

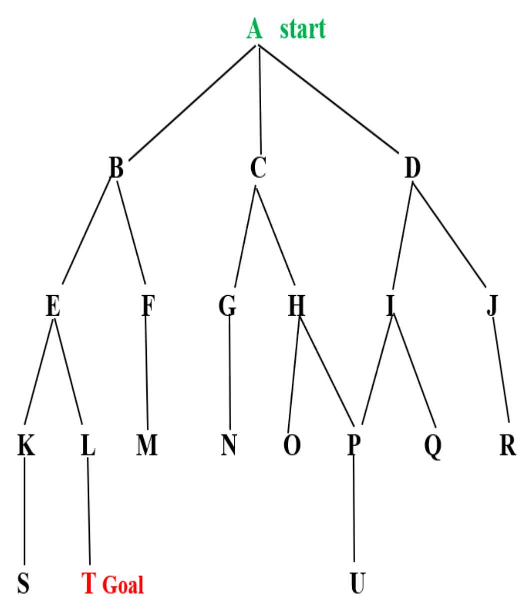
a) Hill Climbing Search.
b) Best First Search.
c) A* Search.
d) Branch and Bound search

Lecture 9: Blind Search (Depth First, Breadth First)

Depth First Search (DFS)

Depth-first search or vertical search Depth-first search When a variable is checked, all its children and grandchildren are checked before any of its siblings. This search starts by checking the island nodes, then selects one of its child nodes and checks it for matching, then the children of the children until it reaches the last depth in the search tree and returns to the parents of the children to check and match other children and so on until it reaches the target state, if any. The search process goes deeper into the search space whenever possible only when there are no other children of the variable. This search process has two advantages: ease of programming, but it is not effective at a deeper level: saving one path and always replacing it when moving from one node to another, as we need a small storage capacity. The difference from the previous mechanism was that it searched at one level for the siblings and when finished, it moved to the other level of the siblings.

Open	Closed
1-open =[A];	Closed = []
2-open = [B, C, D];	Closed = [A]
3-open = [E, F, C, D];	Closed = [B, A]
4-open = [K, L, F, C, D];	Closed = [E, B, A]
5-open = [S, L, F, C, D];	Closed = [K, E, B, A]
6-open = [L, F, C, D];	Closed = [S, K, E, B, A]
7-open = [T, F, C, D];	Closed = [L, S, K, E, B, A]

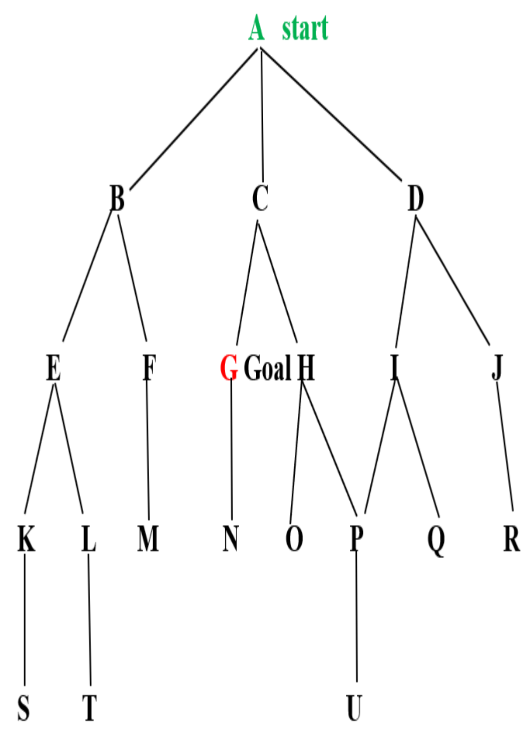


Path = [A, B, E, K, S, L, T]

Breadth First Search (BFS)

Breadth-First Search (BFS)is one of the primary methods used in search processes. In BFS, each node's siblings are examined before its children. Starting from the root node, such as node A, the algorithm first generates all the children of node A (e.g., nodes B and C) and tests them. Then, it proceeds to examine the children of node B, and so on, ensuring that each level is fully explored before moving to the next. This method searches the tree horizontally, level by level, until the last level is reached, beginning from the starting state. BFS relies on two states: "open" (unexplored nodes) and "closed" (already explored nodes).

In the following example, BFS is used to find a path between nodes A and G. The steps of the process are as follows: since the target node is not specifie continue searching until the last node in the tree is reached.



Open	closed
1-open = [A];	Closed = []
2-open = [B, C, D];	Closed = [A]
3-open = [C, D, E, F];	Closed = [A, B]
4-open = [D, E, F, G, H];	Closed = [C, B, A]
5-open = [E, F, G, H, I, J];	Closed = [D, C, B, A]
6-open = [F, G, H, I, J, K, L];	Closed = [E, D, C, B, A]
7-open = [G, H, I, J, K, L, M];	Closed = [F, E, D, C, B, A]

Path = [A, B, C, D, E, F, G].

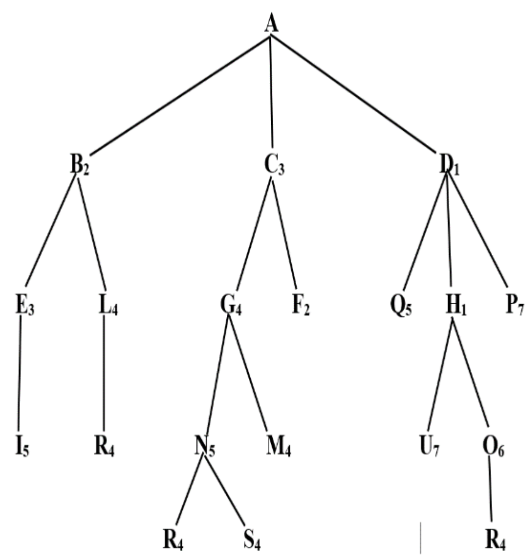
Lecture 10: Heuristic Search (Hill Climb Search, Best First Search)

Hill Climb Search

Hill climbing is a heuristic search technique similar to depth-first search. The simplest way to implement this type of search is using a hill climbing strategy. In this method, the current state is expanded, and its children are evaluated. The best (and often least expensive) child is chosen for further exploration. Unlike other search methods, siblings and parent nodes are not retained. The basic idea is to avoid managing a large list of node variables; instead, the search focuses on ensuring that one option is better than the others and ranking the options accordingly. Based on the footprint of an individual node and the path that led to it from the initial state, you can select the next node that will bring you closer to the goal, according to the estimated heuristic cost, g(n), and continue from there. The term "hill climbing" reflects the process of trying to reach the top of a hill by always moving upward from your current location.

In the following example Hill Climb Search:

Open	Closed	X
1-open = [A]	Closed = []	**A**
2-open = [D_1, B_2, C_3]	Closed = [A]	**D_1**
3-open = [H_1, Q_5, P_7]	Closed = [A, D_1]	**H_1**
4-open = [O_6, U_7]	Closed = [A, D_1, H_1]	**O_6**
5-open = [R_4]	Closed = [A, D_1, H_1, O_6]	**R_4**

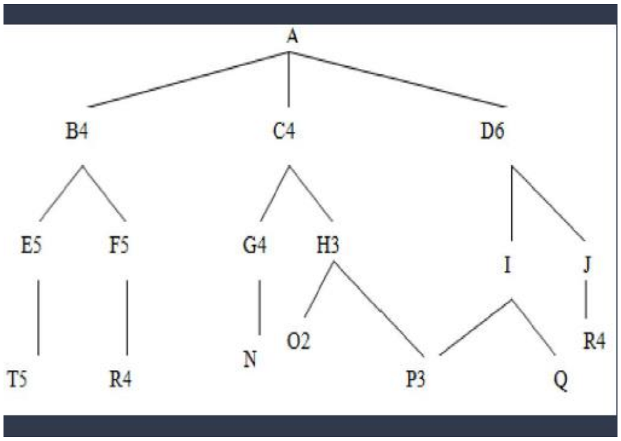


The solution path is: [A, D_1, H_1, O_6, R_4]

Best First Search

In this search, the total costs of each state in the search space are evaluated, with priority given to nodes closest to the goal. In other words, nodes with the lowest costs to reach the goal are selected over others.

In the following example Best First Search:



Open	closed
1. Open=[A5]	Closed=[]
2. Open=[B4,C4,D6]	Closed=[A5]
3. Open=[C4,E5,F5,D6]	Closed=[B4,A5]
4. Open=[H3,G4,E5,F5,D6]	Closed=[C4,B4,A5]
5. Open=[O2,P3,G4,E5,F5,D6]	Closed=[H3,C4,B4,A5]
6. Open=[P3,G4,E5,F5,D6]	Closed=[O2,H3,C4,B4,A5]
7. Open=[G4,E5,F5,D6]	Closed=[P3,O2,H3,C4,B4,A5]

The solution path is: A5 -B4 -C4 -H3 –O2-P3

Lecture 11: Heuristic Search (Branch and Bound search, A* Search)

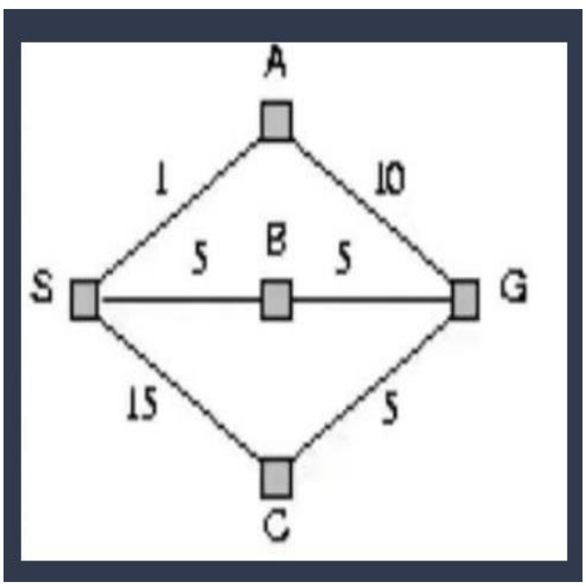
Branch and Bound Search (Uniformed Cost Search)

The Branch and Bound Algorithm is a method used in combinatorial optimization problems to systematically search for the best solution. It works by dividing the problem into smaller subproblems, or branches, and then eliminating certain branches based on bounds on the optimal solution. This process continues until the best solution is found or all branches have been explored. Branch and bound is commonly used in problems like the traveling salesman and job scheduling.

The heuristic function:

F(n) = G(n)

Example:



Open	closed
Open=[s]	closed=[]
Open= [a1,b5,c15]	closed= [s0]
Open= [b5,g11,c15]	closed= [a1]
Open= [g10,c15]	closed= [a1,b5]

The goal is found & the resulted path is: s0 b5 g5 =10]

A* Search

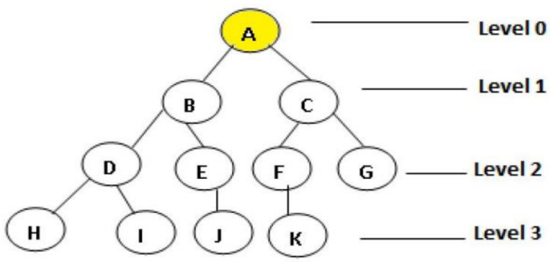
The algorithm of this search is similar to the best-first algorithm, but with a different mining function where f(n) is defined as follows:

F(n) = g(n) + h(n)

Where

h(n): is the sum of the costs from the beginning to n

g(n): is an estimate of the cost of the path from n to the goal

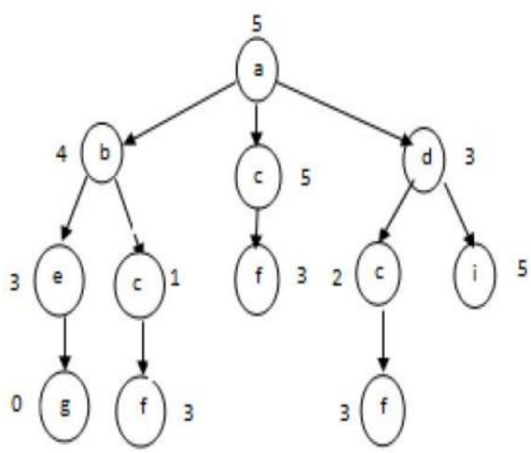


The heuristic function

F (n) = h (n) + g (n)

Example:

Find the solution path of the search tree using A* Search



Open	closed
1. Open=[A5]	closed=[ ]
2. Open=[D4 , B5 , C6]	closed=[A5]
3. Open=[C4 , B5,I7]	closed=[A5 , D4]
4. Open=[B5 , F6,I7]	closed=[A5 , D4 , C4]
5. Open=[C3 , E5 , F6,I7]	closed=[A5 , D4 ,C4, B5]
6. Open=[E5 , F6,I7]	closed=[A5 , D4 , B5 , C3]
7. Open=[G3 , F6,I7]	closed=[A5 , D4 , B5 , C3 , E5]

the resulted path is : A5 ->D3 -> B4 -> C1 -> E3 -> G0 = 16

Lecture 12: Expert Systems

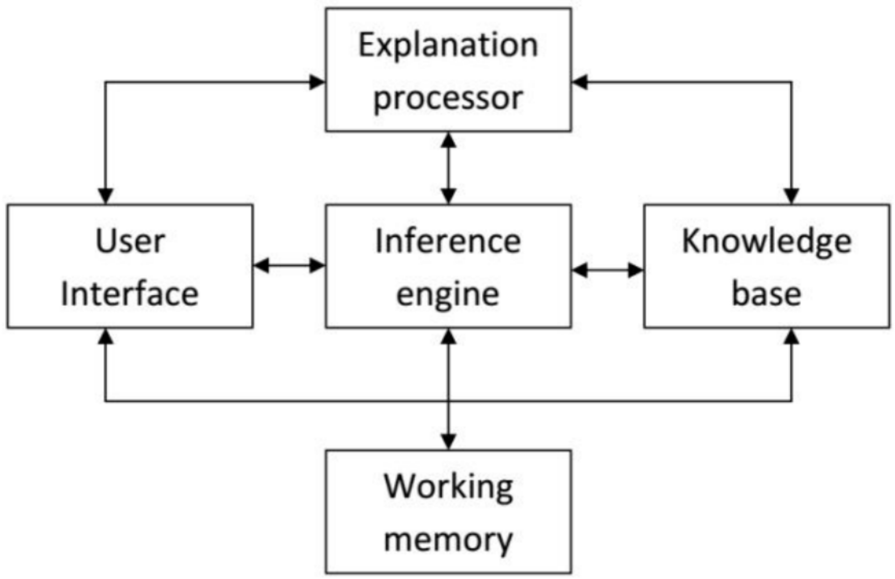
What are expert systems?

Expert systems are computer programs that are constructed to do the kinds of activities that human experts can do such as design, compose, plan, diagnose, interpret, summarize, audit, give advice

Applications of Expert System

Application	Description
Design Domain	Camera lens design, automobile design
Medical Domain	Diagnosis Systems to deduce the cause of disease from observed data, conduction medical operations on humans
Monitoring System	Comparing data continuously with the observed system or with prescribed behavior such as leakage monitoring in a long petroleum pipeline.
Process Control Systems	Controlling a physical process based on monitoring
Knowledge Domain	Finding out faults in vehicles, computers.
Finance/Commerce	Detection of possible fraud, suspicious transactions, stock market trading, Airline scheduling, cargo scheduling.

What is Expert System Architecture and Components?



1- User Interface:

The user interacts with the expert system through a user interface that makes access more comfortable for the human and hides much of the system complexity.

2- Explanation Processor:

The explanation part allows the program to explain its reasoning to the user. These explanations include justifications for the system's conclusion (how queries), and an explanation of why the system needs a particular piece of data (why queries).

3- Knowledge Base:

The heart of the expert system contains the problem-solving knowledge (which is defined as an original collection of processed information) of the particular applications, this knowledge is represented in several ways such as if-then rules form.

4- Inference Engine

The inference engine applies the knowledge to the solution of actual problems. It's the interpreter for the knowledge base. The inference engine performs the recognize act control cycle. To recommend a solution, the Inference Engine uses the following strategies.

- Forward Chaining
- Backward Chaining

5- Working Memory:

It is a part of the memory used for matching rules and calculations. When the work is finished this memory will be raised.

Benefits of Expert Systems

- Ability to simulate human experience:
- Inference engine
- Ability to interpret results
- Flexibility and self-learning
- Dealing with uncertainty
- User interaction
- Specialization
- Speed and efficiency
- Updatable

Advantages VS Disadvantages of An Expert System

Advantages	Disadvantages
• Expensive To Produce • Time Consuming • Lacks Depth • Inflexible • No Common Sense • Programs Have Error	• Increased availability • Reduced cost • Reduced danger • Performance • Multiple expertise • increased

Lecture 13: Limitations in expert systems and some of their applications

Limitations of Expert Systems

Despite the significant benefits offered by expert systems, there are several **limitations** or challenges associated with these systems:

1. Limited Knowledge Base:

Expert systems rely heavily on a **knowledge base**, and if this base is insufficient or outdated, the system may provide inaccurate or inappropriate solutions. The knowledge base needs regular updates to keep pace with developments in the field.

2. Handling Complexity and Ambiguity:

Some expert systems may struggle with **ambiguous** or **changing information**. Although tools like fuzzy logic are available, dealing with situations where information is incomplete or unclear can remain a challenge.

3. Lack of Self-Learning Ability:

Most traditional expert systems rely on the knowledge input into them and cannot **learn or evolve independently** based on new experiences. This contrasts with machine learning systems that can improve their performance over time.

4. High Development Costs:

Developing an expert system requires significant time and effort to gather and organize knowledge. Additionally, experts are needed to properly train the system, making development costs **high**.

5. Inability to Think Creatively:

Expert systems **cannot think creatively or solve problems in new ways**. They are constrained by the rules and guidelines previously input, which makes them unable to adapt to unfamiliar challenges.

6. Dependence on Data Quality:

If the **input data is inaccurate or incomplete**, the expert system will provide inaccurate outputs. The system needs high-quality data to function efficiently.

7. Limited User Interaction:

Some expert systems may be limited in **how they interact with users**, especially if their interfaces are not intuitive or user-friendly.

Applications of Expert Systems

1. Medicine (Medical Diagnosis):

One of the most well-known applications in medicine is **expert systems for disease diagnosis**. An example is the **MYCIN** system, designed to diagnose bacterial infections and recommend treatments. The system relies on a vast medical knowledge base to assist doctors in decision making.

2. Agriculture:

Expert systems can help provide recommendations to farmers on **best agricultural practices**, by analyzing data related to soil, climate, and different plant species.

3. Finance and Accounting:

Expert systems are used in the financial sector to offer advice on **investment management**, market analysis, and economic forecasting. They are also employed for tax compliance monitoring and financial statement analysis.

4. Manufacturing and Production:

In **industry**, expert systems are used to provide solutions for improving production processes, such as scheduling and planning operations, predictive maintenance, and quality management.

5. Artificial Intelligence in Robotics:

- Expert systems are used in **robotics** to interact with the surrounding environment, make real-time decisions, and analyze complex situations.

6. Law:

Expert systems assist in the **legal field** by providing legal advice based on statutes and regulations. They can also help lawyers analyze legal cases and research legal precedents.

7. Education:

Expert systems can be used to develop educational systems that provide **personalized educational advice** for each student based on their needs and educational level, known as "personalized learning" or **intelligent tutoring systems**.