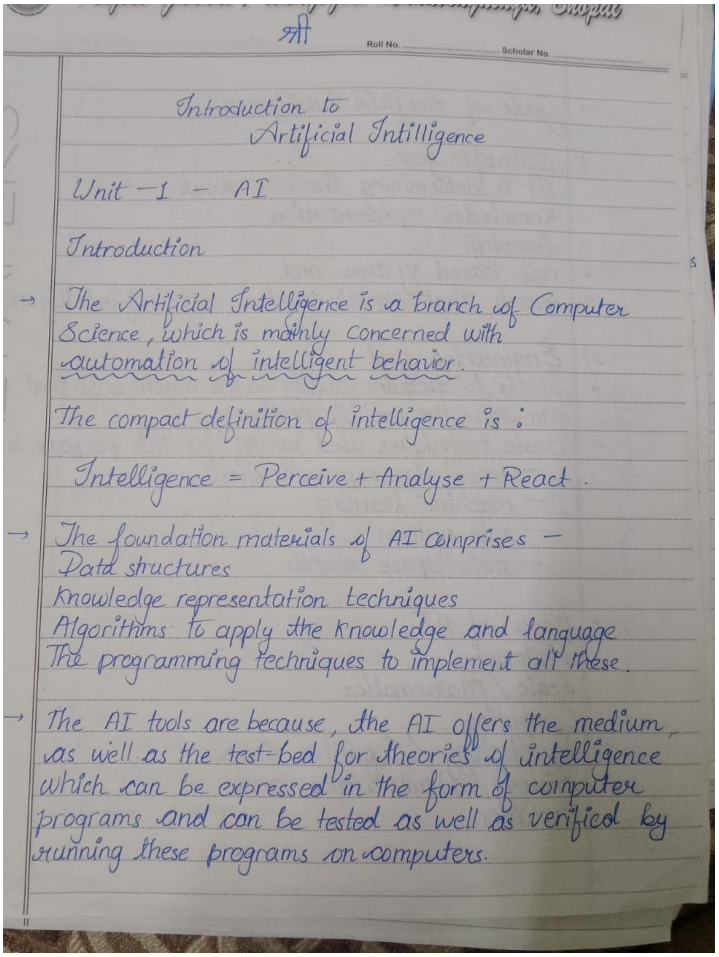
Assignments

AL-205
(Introduction to Artificial Intelligence)

1. Why do we need Artificial Intelligence? Give some real-world applications of Al.



Some examples of AI include:

There are numerous examples of AI applications across various industries. Here are some common examples:

- **Speech recognition:** speech recognition systems use deep learning algorithms to recognize and classify images and speech. These systems are used in a variety of applications, such as self-driving cars, security systems, and medical imaging.
- **Personalized recommendations:** E-commerce sites and streaming services like Amazon and Netflix use AI algorithms to analyze users' browsing and viewing history to recommend products and content that they are likely to be interested in.

- **Predictive maintenance:** AI-powered predictive maintenance systems analyze data from sensors and other sources to predict when equipment is likely to fail, helping to reduce downtime and maintenance costs.
- **Predictive analytics –** AI is used in industries such as healthcare and marketing to analyze large amounts of data and make predictions about future events, such as disease outbreaks or consumer behavior.
- **Medical diagnosis:** AI-powered medical diagnosis systems analyze medical images and other patient data to help doctors make more accurate diagnoses and treatment plans.
- **Autonomous vehicles:** Self-driving cars and other autonomous vehicles use AI algorithms and sensors to analyze their environment and make decisions about speed, direction, and other factors.
- **Virtual Personal Assistants (VPA) like Siri or Alexa –** these use natural language processing to understand and respond to user requests, such as playing music, setting reminders, and answering questions.
- **Fraud detection –** financial institutions use AI to analyze transactions and detect patterns that are indicative of fraud, such as unusual spending patterns or transactions from unfamiliar locations.
- **Image recognition –** AI is used in applications such as photo organization, security systems, and autonomous robots to identify objects, people, and scenes in images.
- **Natural language processing –** AI is used in chatbots and language translation systems to understand and generate human-like text.
- **Game-playing AI –** AI algorithms have been developed to play games such as chess, Go, and poker at a superhuman level, by analyzing game data and making predictions about the outcomes of moves.

---

2. How Artificial intelligence, Machine Learning, and Deep Learning differ from each other?

<u>Artificial Intelligence</u>

Artificial Intelligence is basically the mechanism to incorporate human intelligence into machines through a set of rules(algorithm). AI is a combination of two words: “Artificial” meaning something made by humans or non-natural things and “Intelligence” meaning the

ability to understand or think accordingly. Another definition could be that **"AI is basically the study of training your machine(computers) to mimic a human brain and its thinking capabilities"**.

AI focuses on 3 major aspects(skills): learning, reasoning, and self-correction to obtain the maximum efficiency possible.

Machine Learning:

Machine Learning is basically the study/process which provides the system(computer) to learn automatically on its own through experiences it had and improve accordingly without being explicitly programmed. **ML is an application or subset of AI.** ML focuses on the development of programs so that it can access data to use it for itself. The entire process makes observations on data to identify the possible patterns being formed and make better future decisions as per the examples provided to them. **The major aim of ML is to allow the systems to learn by themselves through experience without any kind of human intervention or assistance.**

Deep Learning:

Deep Learning is basically a sub-part of the broader family of Machine Learning which makes use of **Neural Networks**(similar to the neurons working in our brain) to mimic human brain-like behavior. DL algorithms focus on **information processing patterns** mechanism to possibly identify the patterns just like our human brain does and classifies the information accordingly. DL works on larger sets of data when compared to ML and the **prediction mechanism is self-administered by machines**.

Below is a table of differences between Artificial Intelligence, Machine Learning and Deep Learning:

Parameters	Artificial Intelligence	Machine Learning	Deep Learning
Definition	AI refers to the broad field of computer science that focuses on creating intelligent machines that can perform tasks that	ML is a disciple that focuses on developing algorithms that can learn from data and improve their performance over time	DL is a disciple that focuses on developing deep neural networks that can automatically learn and extract

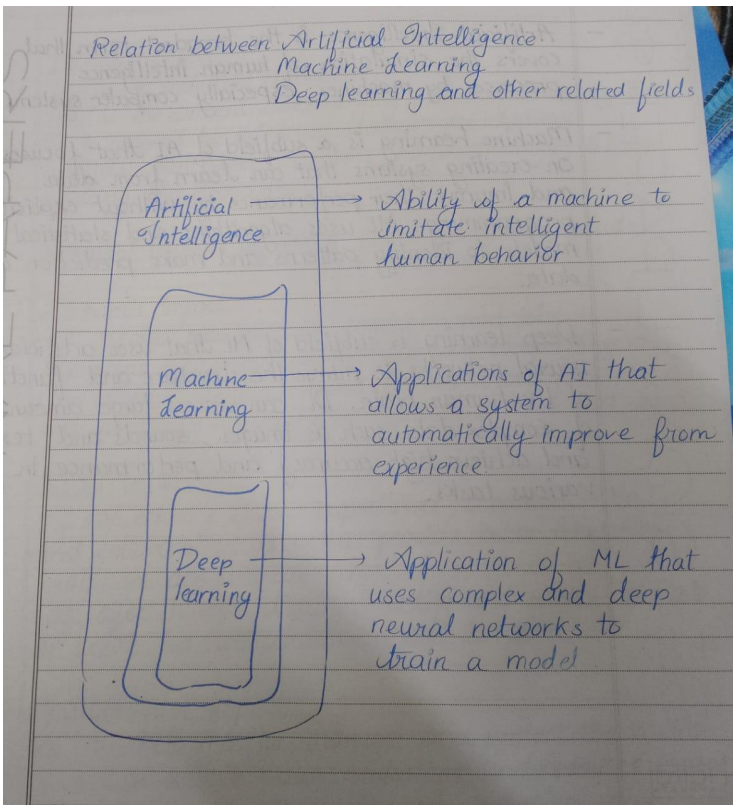
	would normally require human intelligence, such as reasoning, perception, and decision-making.	without being explicitly programmed.	features from data.
Subset of or components	AI is the broader family consisting of ML and DL as it's components.	ML is the subset of AI.	DL is the subset of ML.
Procedure	AI is a computer algorithm which exhibits intelligence through decision making.	ML is an AI algorithm which allows system to learn from data.	DL is a ML algorithm that uses deep(more than one layer) neural networks to analyze data and provide output accordingly.

Maths involved	Search Trees and much complex math is involved in AI.	If you have a clear idea about the logic(math) involved in behind and you can visualize the complex functionalities like K-Mean, Support Vector Machines, etc., then it defines the ML aspect.	If you are clear about the math involved in it but don't have idea about the features, so you break the complex functionalities into linear/lower dimension features by adding more layers, then it defines the DL aspect.
Aim	The aim is to basically increase chances of success and not accuracy.	The aim is to increase accuracy not caring much about the success ratio.	It attains the highest rank in terms of accuracy when it is trained with large amount of data.

Categories	Three broad categories/types Of AI are: Artificial Narrow Intelligence (ANI), Artificial General Intelligence (AGI) and Artificial Super Intelligence (ASI)	Three broad categories/types Of ML are: Supervised Learning, Unsupervised Learning and Reinforcement Learning	DL can be considered as neural networks with a large number of parameters layers lying in one of the four fundamental network architectures: Unsupervised Pre-trained Networks, Convolutional Neural Networks, Recurrent Neural Networks and Recursive Neural Networks
Efficiency	The efficiency Of AI is basically the efficiency provided by ML and DL respectively.	Less efficient than DL as it can't work for longer dimensions or higher amount of data.	More powerful than ML as it can easily work for larger sets of data.

Examples	Examples of AI applications include: Google's AI-Powered Predictions, Ridesharing Apps Like Uber and Lyft, Commercial Flights Use an AI Autopilot, etc.	Examples of ML applications include: Virtual Personal Assistants: Siri, Alexa, Google, etc., Email Spam and Malware Filtering.	Examples of DL applications include: Sentiment based news aggregation, Image analysis and caption generation, etc.
Sub fields	AI can be further broken down into various subfields such as robotics, natural language processing, computer vision, expert systems, and more.	ML algorithms can be categorized as supervised, unsupervised, or reinforcement learning. In supervised learning, the algorithm is trained on labeled data, where the desired output is known. In unsupervised learning, the algorithm is trained on	DL algorithms are inspired by the structure and function of the human brain, and they are particularly well-suited to tasks such as image and speech recognition.

		unlabeled data, where the desired output is unknown.	
Sub Methods	AI systems can be rule-based, knowledge-based, or data-driven.	In reinforcement learning, the algorithm learns by trial and error, receiving feedback in the form of rewards or punishments.	DL networks consist of multiple layers of interconnected neurons that process data in a hierarchical manner, allowing them to learn increasingly complex representations of the data.



---

3. Discuss the impact of artificial intelligence on industries:

Artificial intelligence (AI) has emerged as a transformative technology with the potential to revolutionize various industries. With its ability to analyze vast amounts of data, learn from patterns, and make intelligent decisions, AI is reshaping the way businesses operate and opening up new opportunities for innovation. In this article, we will explore the impact of artificial intelligence on various industries and how it is driving significant advancements.

1. **Healthcare:**

   AI is revolutionizing the healthcare industry by enabling more accurate diagnoses, personalized treatments, and improved patient outcomes. Machine learning algorithms can analyze medical images, such as X-rays and MRIs, to detect anomalies and assist radiologists in their diagnoses. AI-powered chatbots and virtual assistants are enhancing patient interactions and providing 24/7 support. Additionally, AI-driven predictive analytics can help healthcare providers identify high-risk patients and intervene early to prevent diseases.

2. **Finance:**

   AI is transforming the finance industry by automating repetitive tasks, improving fraud detection, and enhancing customer experiences. AI algorithms can analyze vast amounts of financial data, identify patterns, and make real-time predictions for investment decisions. Robo-advisors are using AI to provide personalized financial advice to investors. AI-powered chatbots and virtual assistants are streamlining

customer service, addressing inquiries, and simplifying banking processes. Moreover, AI-based algorithms can detect fraudulent transactions and prevent financial crimes more effectively.

3. **Manufacturing:**

AI is driving the concept of smart manufacturing by optimizing production processes, reducing downtime, and improving product quality. AI-powered robots and automation systems are taking over repetitive tasks, leading to increased efficiency and productivity. Machine learning algorithms are used to predict maintenance needs and schedule proactive repairs, minimizing unplanned downtime. AI-enabled quality control systems can detect defects in real-time, ensuring products meet the highest standards.

4. **Retail:**

AI is revolutionizing the retail industry by enhancing customer experiences, improving inventory management, and enabling personalized marketing. Recommendation systems powered by AI algorithms provide personalized product suggestions based on customer preferences and browsing history. Chatbots and virtual assistants can assist customers in finding products, answering queries, and providing support. AI-driven analytics help retailers optimize inventory levels, forecast demand, and reduce waste. Furthermore, computer vision technology is enabling cashier-less stores, creating frictionless shopping experiences.

5. **Transportation**:

AI is making significant advancements in the transportation industry, particularly in autonomous vehicles and logistics optimization. AI algorithms power self-driving cars, enabling them to navigate roads, detect obstacles, and make real-time decisions. This technology has the potential to enhance safety, reduce traffic congestion, and improve fuel efficiency. AI-based algorithms optimize logistics and supply chain management, enabling efficient route planning, predictive maintenance, and demand forecasting.

6. **Education:**

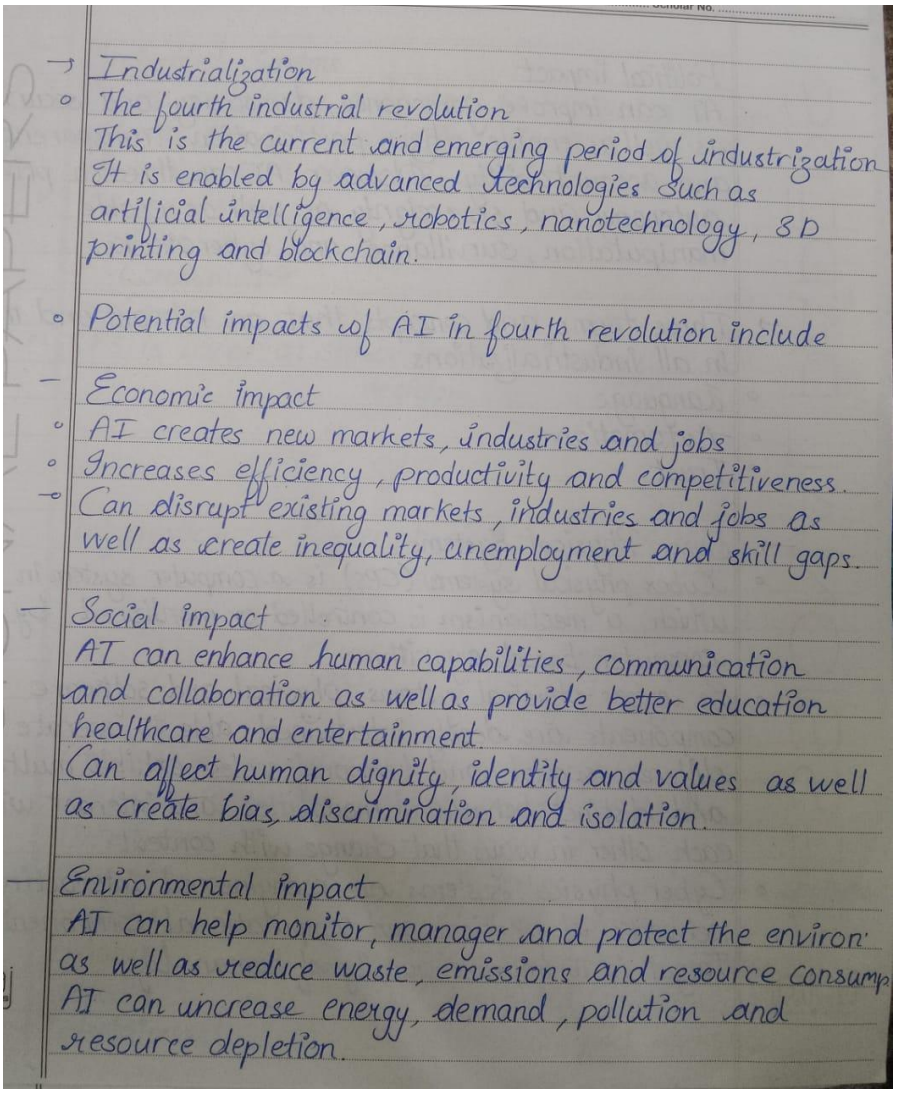
AI is transforming the education industry by personalizing learning experiences, automating administrative tasks, and facilitating adaptive assessments. AI-powered learning platforms can adapt content based on individual student needs, providing personalized education pathways. Intelligent tutoring systems use AI algorithms to provide tailored feedback and support to students. AI-enabled chatbots are automating administrative tasks, such as answering student inquiries and managing schedules, freeing up educators' time for more meaningful interactions.

7. **Agriculture:**

AI is revolutionizing the agricultural sector by optimizing crop yields, reducing resource consumption, and improving farming practices. AI algorithms analyze environmental data, soil conditions, and weather patterns to provide farmers with

insights on optimal planting times, irrigation schedules, and pest control measures. AI-powered drones and robots can monitor crops, identify disease outbreaks, and automate labor-intensive tasks. This technology helps increase productivity, reduce costs, and promote sustainable farming practices.

In conclusion, artificial intelligence is having a profound impact on various industries, transforming the way businesses operate and delivering significant advancements. From healthcare and finance to manufacturing and transportation, AI is driving innovation, improving efficiency, and enhancing customer experiences.



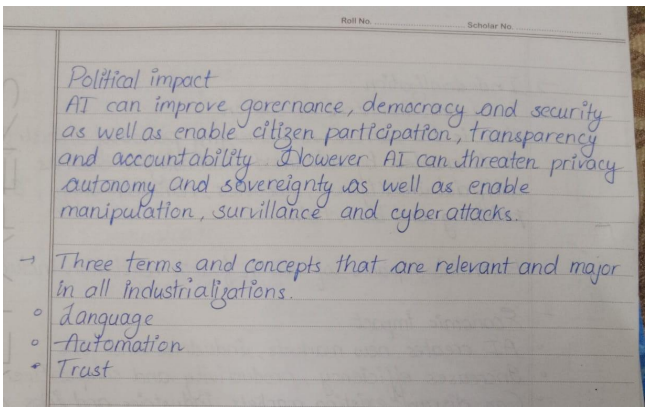
→ Industrialization
- The fourth industrial revolution
This is the current and emerging period of industrization
It is enabled by advanced technologies such as artificial intelligence, robotics, nanotechnology, 3D printing and blockchain.

- Potential impacts of AI in fourth revolution include

— Economic impact
- AI creates new markets, industries and jobs
- Increases efficiency, productivity and competitiveness.
- Can disrupt existing markets, industries and jobs as well as create inequality, unemployment and skill gaps.

— Social impact
AI can enhance human capabilities, communication and collaboration as well as provide better education healthcare and entertainment.
Can affect human dignity, identity and values as well as create bias, discrimination and isolation.

— Environmental impact
AI can help monitor, manager and protect the environ
as well as reduce waste, emissions and resource consump
AI can increase energy, demand, pollution and resource depletion.

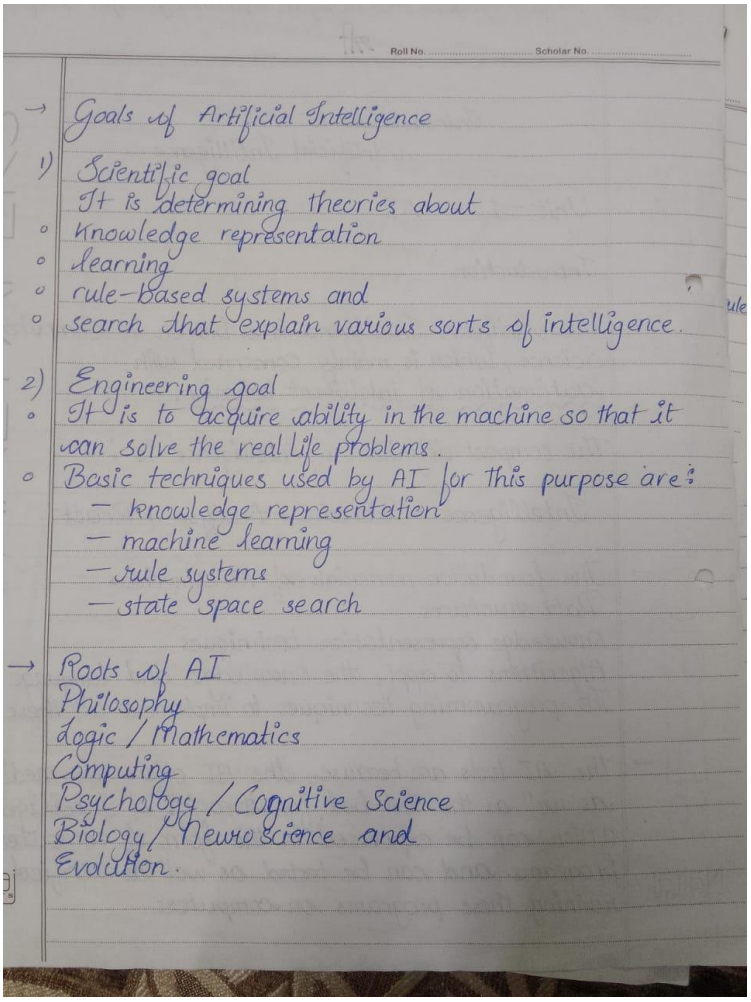


Political impact
AI can improve gorernance, democracy and security as well as enable citizen participation, transparency and accountability. However AI can threaten privacy autonomy and sovereignty as well as enable manipulation, surviillance and cyber attacks.

→ Three terms and concepts that are relevant and major in all Industrializations.
- Language
- Automation
- Trust

---

4. Discuss the evolution of Artificial intelligence. Mention its primary goals also



→ Goals of Artificial Intelligence

1) Scientific goal
It is determining theories about
- knowledge representation
- learning
- rule-based systems and
- search that explain various sorts of intelligence.

2) Engineering goal
- It is to acquire ability in the machine so that it can solve the real life problems.
- Basic techniques used by AI for this purpose are:
  - knowledge representation
  - machine learning
  - rule systems
  - state space search

→ Roots of AI
Philosophy
Logic / Mathematics
Computing
Psychology / Cognitive Science
Biology / Neuro Science and
Evolution.

---

5. Which Assessment is used to Test the Intelligence of a Machine? Explain It.

- The Turing Test, proposed by Alan Turing (1950), was designed to provide a satisfactory operational definition of intelligence.

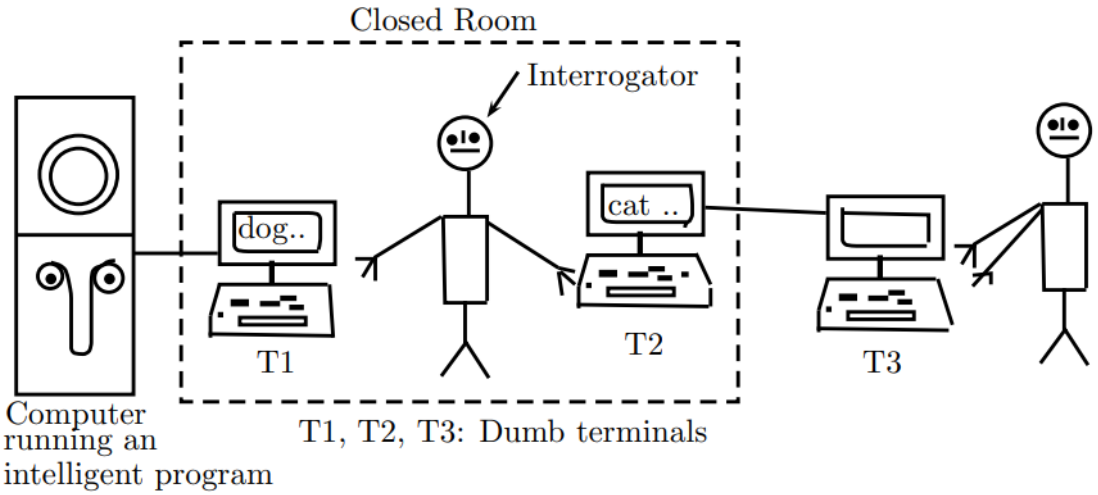
- A computer passes the test if a human interrogator, after posing some written questions, cannot tell whether the written responses come from a person or from a computer.
- The computer would need to possess the following capabilities:

1. **natural language processing** to enable it to communicate successfully in English;
2. **knowledge representation** to store what it knows or hears;
3. **automated reasoning** to use the stored information to answer questions and to draw new conclusions;
4. **machine learning** to adapt to new circumstances and to detect and extrapolate patterns.

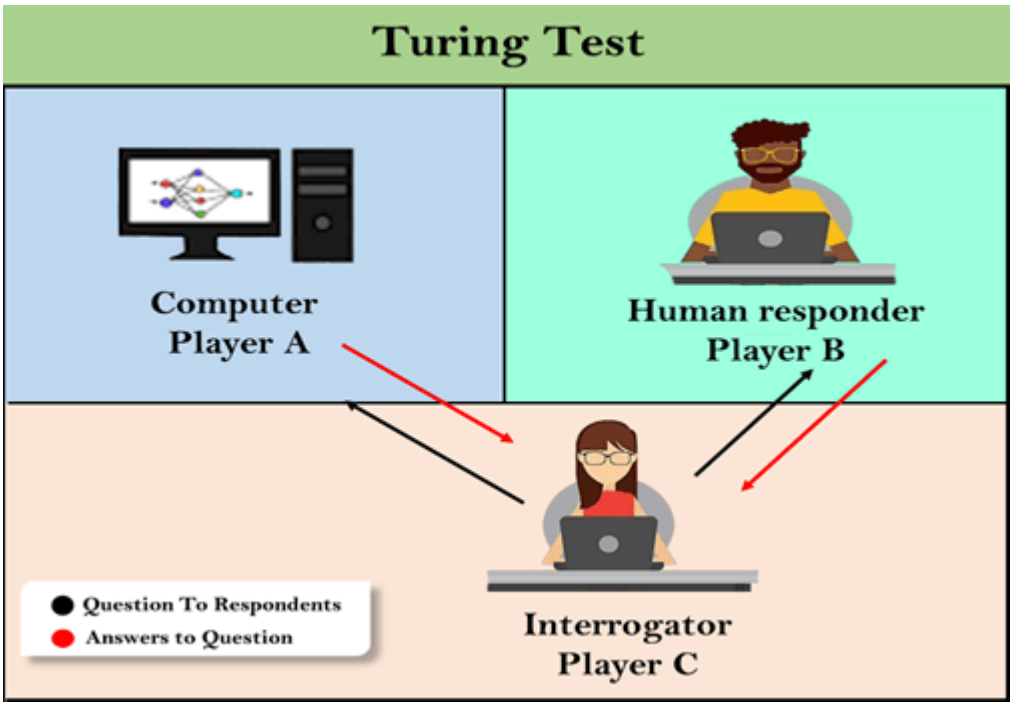
- Turing's test deliberately avoided direct physical interaction between the interrogator and the computer, because physical simulation of a person is unnecessary for intelligence.
- To pass the **total Turing Test**, the computer will need
    a. **computer vision** to perceive objects, and
    b. **robotics** to manipulate objects and move about

Process:

- In 1950, in an article "Computing Machinery and Intelligence," Alan M. Turing proposed an empirical test for machine intelligence, now called **Turing Test**.
- It is designed to measure the performance of an intelligent machine against humans, for its intelligent behavior.
- Turing called it imitation game, where machine and human counter-part are put in different rooms, separate from a third person, called interrogator.
- The interrogator is not able to see or speak directly to any of the other two, and does not know which entity is a machine, and communicates to these two solely by textual devices like a dumb terminal.
- The interrogator is supposed to distinguish the machine from the human solely based on the answers received for the questions asked over the interface device, which is a keyboard (or teletype).
- Even after having asked the number of questions, if the interrogator is not able to distinguish the machine from the human, then as per the argument of Alan Turing, the machine can be considered intelligent.
- Interrogator may ask highly computation oriented questions to identify the machine, and other questions related to general awareness, poetry, etc., to identify the human.



Turing test (imitation game)



The Chinese Room Argument:

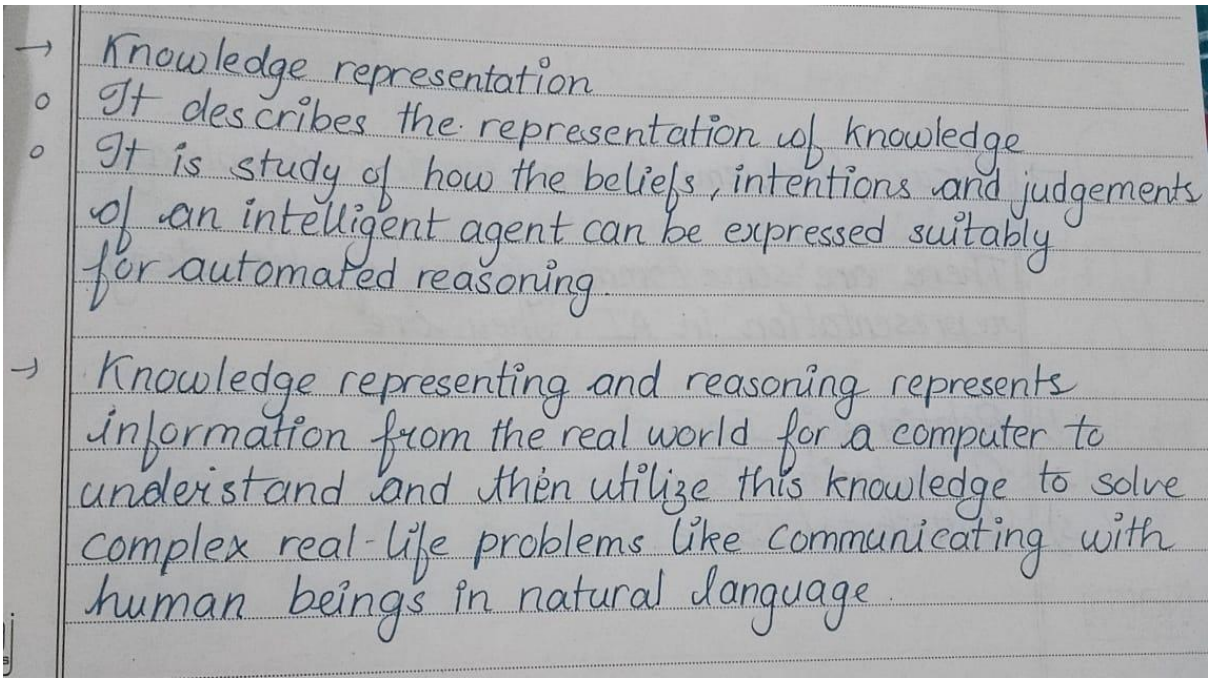
There were many philosophers who really disagreed with the complete concept of Artificial Intelligence. The most famous argument in this list was "**Chinese Room**."

In the year **1980, John Searle** presented "**Chinese Room**" thought experiment, in his paper "**Mind, Brains, and Program**," which was against the validity of Turing's Test. According to his argument, "**Programming a computer may make it to understand a language, but it will not produce a real understanding of language or consciousness in a computer**."

He argued that Machine such as ELIZA and Parry could easily pass the Turing test by manipulating keywords and symbol, but they had no real understanding of language. So it cannot be described as "thinking" capability of a machine such as a human.

---

6. Briefly describe the meaning of knowledge representation and knowledge acquisition.
What procedure is followed for knowledge acquisition? Explain.



In AI, knowledge acquisition is the process of extracting knowledge from data. This can be done manually, through a process of observation and experimentation, or automatically, using a variety of techniques such as machine learning.

- In artificial intelligence, knowledge acquisition is the process of gathering, selecting, and interpreting information and experiences to create and maintain knowledge within a specific domain. It is a key component of machine learning and knowledge-based systems.
- There are many different methods of knowledge acquisition, including rule-based systems, decision trees, artificial neural networks, and fuzzy logic systems. The most appropriate method for a given application depends on the nature of the problem and the type of data available.
- The most important part of knowledge acquisition is the interpretation of information. This is where human expertise is required. Machines are not able to interpret information in the same way humans can. They can only make sense of data if it is presented in a certain way.
- Humans need to select the right data and experiences to create knowledge. They also need to interpret that data correctly. This is where artificial intelligence can help. AI

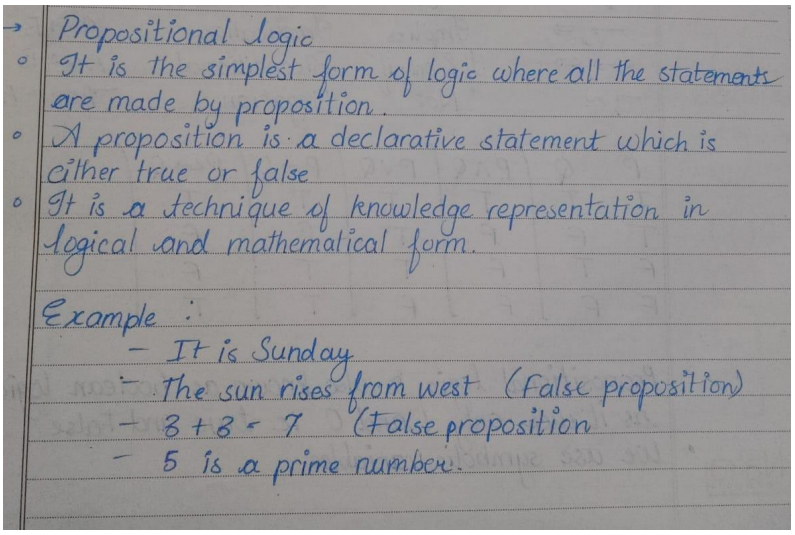
systems can automate the process of knowledge acquisition, making it faster and more accurate.

There are a few methods of knowledge acquisition in AI:
1. **Expert systems**: In this method, experts in a particular field provide rules and knowledge to a computer system, which can then be used to make decisions or solve problems in that domain.
2. **Learning from examples**: This is a common method used in machine learning, where a system is presented with a set of training data, and it "learns" from these examples to generalize to new data.
3. **Natural language processing**: This is a method of extracting knowledge from text data, using techniques like text mining and information extraction.
4. ***Semantic web***: The semantic web is a way of representing knowledge on the internet using standards like RDF and OWL, which can be processed by computers.
5. **Knowledge representation and reasoning:** This is a method of representing knowledge in a formal way, using logic or other formalisms, which can then be used for automated reasoning.

---

7. Define the term logic. Explain its types also.

Logic is one of the essential ideas in the field of artificial intelligence. AI systems' reasoning and decision-making are based on logic. It offers a method for presenting and handling data in a way that enables an AI system to draw conclusions and inferences from it.
Propositional logic



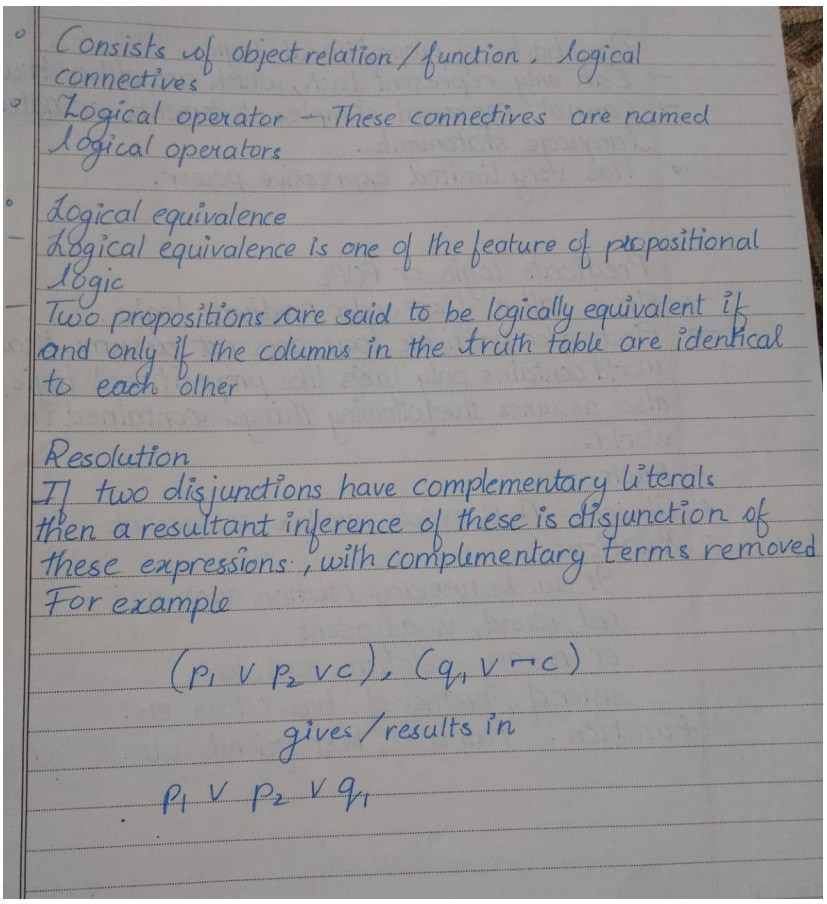
Roll No. ..... Scholar No. .....

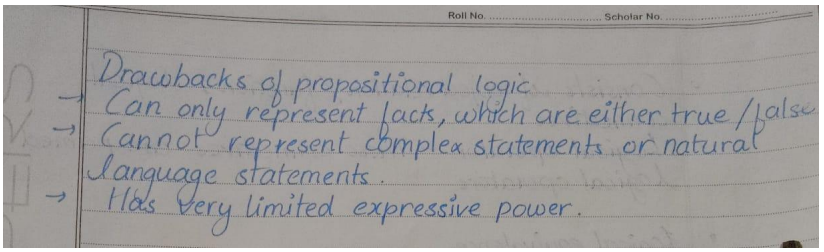
- A proposition formula which always false is called Contradiction.
- A propositional formula which is always true is called tautology, it is also called a valid sentence.
- A proposition formula which has both true and false value is called Contingency.

Connective Symbols	Word	term	Example
$\wedge$	AND	Conjeinction	$A \wedge B$
$\vee$	OR	Disjunction	$A \vee B$
$\rightarrow, \Rightarrow$	Implies	Implication	$A \rightarrow B$
$\leftrightarrow, \Leftrightarrow$	If & only if	Biconditional	$A \Leftrightarrow B$
$\neg, \sim$	Not	Negation	$\neg A$ or $\sim A$

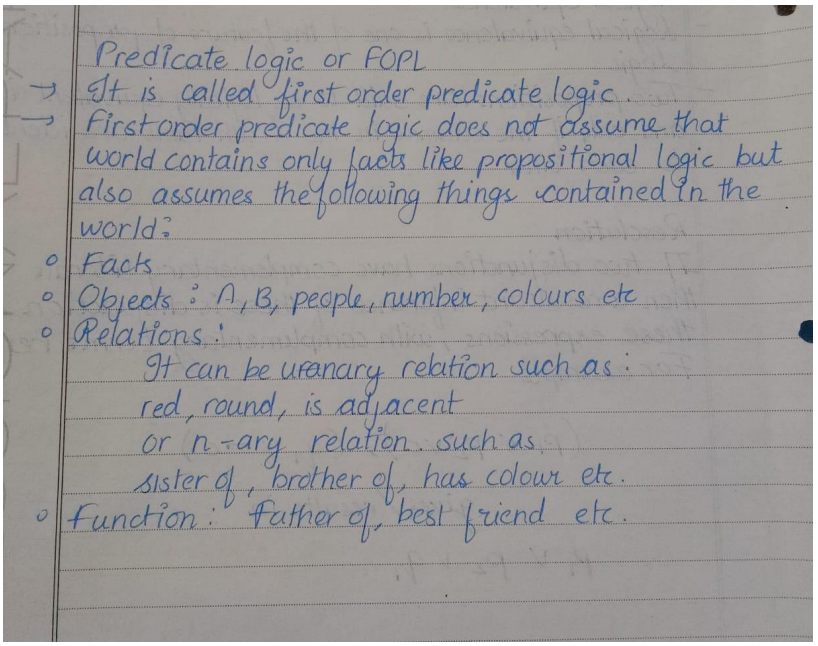
P	Q	$P \wedge Q$	$P \vee Q$	$P \rightarrow Q$	$P \leftrightarrow Q$
T	T	T	T	T	T
T	F	F	T	F	F
F	T	F	T	T	F
F	F	F	F	T	T

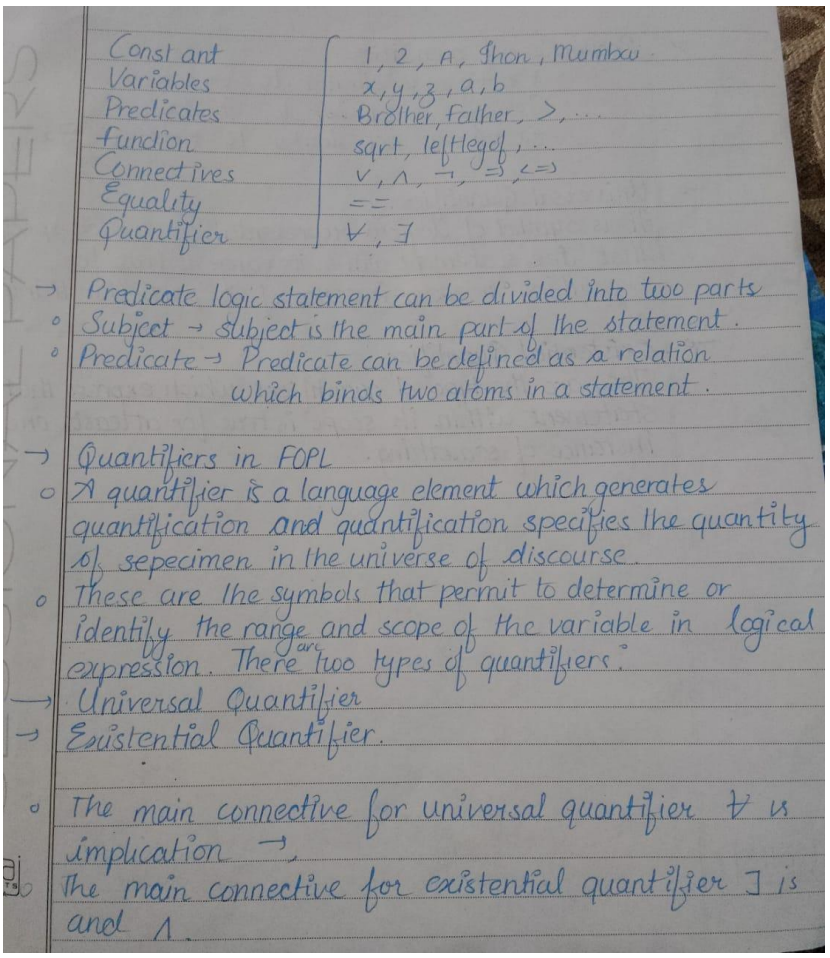
- Propositional logic is also known as boolean logic as it uses only 1 and 0 ie true and false.
- We use symbolic variables.





Predicate logic





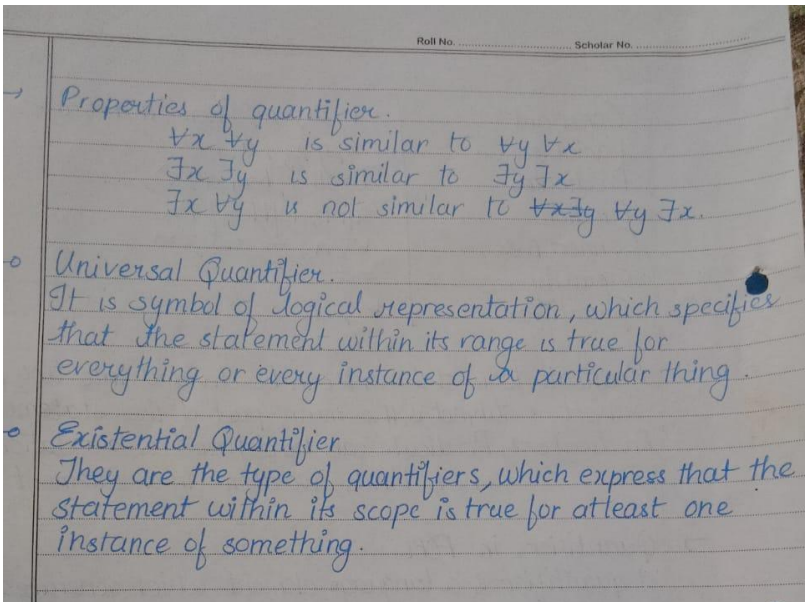
Constant	1, 2, A, Thon, Mumbai
Variables	x, y, z, a, b
Predicates	Brother, father, >, ...
Function	sqrt, leftleg of, ...
Connectives	$\vee$, $\wedge$, $\neg$, $\Rightarrow$, $\Leftrightarrow$
Equality	==
Quantifier	$\forall$, $\exists$

→ Predicate logic statement can be divided into two parts
- Subject → Subject is the main part of the statement.
- Predicate → Predicate can be defined as a relation which binds two atoms in a statement.

→ Quantifiers in FOPL
- A quantifier is a language element which generates quantification and quantification specifies the quantity of sepecimen in the universe of discourse.
- These are the symbols that permit to determine or identify the range and scope of the variable in logical expression. There are two types of quantifiers.

→ Universal Quantifier
→ Existential Quantifier.

- The main connective for universal quantifier $\forall$ is implication →.
- The main connective for existential quantifier $\exists$ is and $\wedge$.



Roll No. ..... Scholar No. .....

→ Properties of quantifier.

$\forall x \forall y$ is similar to $\forall y \forall x$

$\exists x \exists y$ is similar to $\exists y \exists x$

$\exists x \forall y$ is not similar to ~~$\forall x \exists y$~~ $\forall y \exists x$.

- Universal Quantifier.
It is symbol of logical representation, which specifies that the statement within its range is true for everything or every instance of a particular thing.

- Existential Quantifier
They are the type of quantifiers, which express that the statement within its scope is true for atleast one instance of something.

Difference between Propositional Logic and Predicate Logic :

	Propositional Logic	Predicate Logic
1	Propositional logic is the logic that deals with a collection of declarative statements which have a truth value, true or false.	Predicate logic is an expression consisting of variables with a specified domain. It consists of objects, relations and functions between the objects.
2	It is the basic and most widely used logic. Also known as Boolean logic.	It is an extension of propositional logic covering predicates and quantification.
3	A proposition has a specific truth value, either true or false.	A predicate's truth value depends on the variables' value.
4	Scope analysis is not done in propositional logic.	Predicate logic helps analyze the scope of the subject over the predicate. There are three quantifiers : Universal Quantifier (∀) depicts for all, Existential Quantifier (∃) depicting there exists some and

		Uniqueness Quantifier (∃!) depicting exactly one.
5	Propositions are combined with Logical Operators or Logical Connectives like Negation(¬), Disjunction(∨), Conjunction(∧), Exclusive OR(⊕), Implication(⇒), Bi-Conditional or Double Implication(⇔).	Predicate Logic adds by introducing quantifiers to the existing proposition.
6	It is a more generalized representation.	It is a more specialized representation.
7	It cannot deal with sets of entities.	It can deal with set of entities with the help of quantifiers.

---

8. Prove that the following sentence is valid.
"If price fall then sell increases. If sell increases then John makes the whole money. But John doesn't make the whole money. Therefore, price do not fall"

---

9. Explain inference rules with example.

Inference:

In artificial intelligence, we need intelligent computers which can create new logic from old logic or by evidence, so generating the conclusions from evidence and facts is termed as Inference.

Inference rules:

Inference rules are the templates for generating valid arguments. Inference rules are applied to derive proofs in artificial intelligence, and the proof is a sequence of the conclusion that leads to the desired goal.

In inference rules, the implication among all the connectives plays an important role. Following are some terminologies related to inference rules:

- **Implication: It is one of the logical connectives which can be represented as P → Q. It is a Boolean expression.**
- **Converse: The converse of implication, which means the right-hand side proposition goes to the left-hand side and vice-versa. It can be written as Q → P.**
- **Contrapositive: The negation of converse is termed as contrapositive, and it can be represented as ¬ Q → ¬ P.**
- **Inverse: The negation of implication is called inverse. It can be represented as ¬ P → ¬ Q.**

From the above term some of the compound statements are equivalent to each other, which we can prove using truth table:

P	Q	P → Q	Q→ P	¬ Q → ¬ P	¬ P → ¬ Q.
T	T	T	T	T	T
T	F	F	T	F	T
F	T	T	F	T	F
F	F	T	T	T	T

Hence from the above truth table, we can prove that P → Q is equivalent to ¬ Q → ¬ P, and Q→ P is equivalent to ¬ P → ¬ Q.

Types of Inference rules:

1. Modus Ponens:

The Modus Ponens rule is one of the most important rules of inference, and it states that if P and P → Q is true, then we can infer that Q will be true. It can be represented as:

Notation for Modus ponens: $\frac{P \rightarrow Q, \quad P}{\therefore Q}$

Example:

Statement-1: "If I am sleepy then I go to bed" ==> P→ Q
Statement-2: "I am sleepy" ==> P
Conclusion: "I go to bed." ==> Q.
Hence, we can say that, if P→ Q is true and P is true then Q will be true.

Proof by Truth table:

P	Q	$P \rightarrow Q$
0	0	0
0	1	1
1	0	0
1	1	1

2. Modus Tollens:

The Modus Tollens rule state that if P→ Q is true and ¬ Q is true, then ¬ P will also true. It can be represented as:

Notation for Modus Tollens: $\frac{P \rightarrow Q, \ \sim Q}{\sim P}$

Statement-1: "If I am sleepy then I go to bed" ==> P→ Q
Statement-2: "I do not go to the bed."==> ~Q
Statement-3: Which infers that "I am not sleepy" => ~P

Proof by Truth table:

P	Q	$\sim P$	$\sim Q$	$P \rightarrow Q$
0	0	1	1	1
0	1	1	0	1
1	0	0	1	0
1	1	0	0	1

3. Hypothetical Syllogism:

The Hypothetical Syllogism rule state that if P→R is true whenever P→Q is true, and Q→R is true. It can be represented as the following notation:

Example:

Statement-1: If you have my home key then you can unlock my home. P→Q
Statement-2: If you can unlock my home then you can take my money. Q→R
Conclusion: If you have my home key then you can take my money. P→R

Proof by truth table:

P	Q	R	$P \rightarrow Q$	$Q \rightarrow R$	$P \rightarrow R$
0	0	0	1	1	1
0	0	1	1	1	1
0	1	0	1	0	1
0	1	1	1	1	1
1	0	0	0	1	1
1	0	1	0	1	1
1	1	0	1	0	0
1	1	1	1	1	1

4. Disjunctive Syllogism:

The Disjunctive syllogism rule state that if P∨Q is true, and ¬P is true, then Q will be true. It can be represented as:

Notation of Disjunctive syllogism: $\frac{P \vee Q, \ \neg P}{Q}$

Example:

Statement-1: Today is Sunday or Monday. ==>P∨Q
Statement-2: Today is not Sunday. ==> ¬P
Conclusion: Today is Monday. ==> Q

Proof by truth-table:

P	Q	$\neg P$	$P \vee Q$
0	0	1	0
0	1	1	1
1	0	0	1
1	1	0	1

5. Addition:

The Addition rule is one the common inference rule, and it states that If P is true, then P∨Q will be true.

Notation of Addition: $\frac{P}{P \vee Q}$

Example:

Statement: I have a vanilla ice-cream. ==> P
Statement-2: I have Chocolate ice-cream.
Conclusion: I have vanilla or chocolate ice-cream. ==> (P∨Q)

Proof by Truth-Table:

P	Q	$P \vee Q$
0	0	0
1	0	1
0	1	1
1	1	1

6. Simplification:

The simplification rule state that if P∧ Q is true, then Q or P will also be true. It can be represented as:

Notation of Simplification rule: $\frac{P \wedge Q}{Q}$ Or $\frac{P \wedge Q}{P}$

Proof by Truth-Table:

P	Q	$P \wedge Q$
0	0	0
1	0	0
0	1	0
1	1	1

7. Resolution:

The Resolution rule state that if P∨Q and ¬ P∧R is true, then Q∨R will also be true. It can be represented as

Notation of Resolution $\frac{P \vee Q,\ \neg P \wedge R}{Q \vee R}$

Proof by Truth-Table:

P	¬ P	Q	R	$P \vee Q$	¬ P∧R	$Q \vee R$
0	1	0	0	0	0	0
0	1	0	1	0	0	1
0	1	1	0	1	1	1
0	1	1	1	1	1	1
1	0	0	0	1	0	0
1	0	0	1	1	0	1
1	0	1	0	1	0	1
1	0	1	1	1	0	1

10. Explain resolution in propositional logic and predicate logic

resolution in propositional logic

The resolution rule is an inference which uses deduction approach. It is used in theorem proving. If two disjunctions have complementary literals, then a resultant inference of these is disjunction of these expressions, with complementary terms removed.

If p = p1 $\vee$ p2 $\vee$ c and q = q1 $\vee$ $\neg$c are two formulas, then resolution of p and q results to dropping of c and $\neg$c and disjunction is performed of the remaining propositions of p and q, as follows:

(p1 $\vee$ p2 $\vee$ c), (q1 $\vee$ $\neg$c) p1 $\vee$ p2 $\vee$ q1 (2.2)

The necessary condition for the above is that C should not be a function of any of the p1, p2, q1.

Example : Show by resolution that (p $\rightarrow$ q) $\rightarrow$ [(r $\wedge$ p) $\rightarrow$ (r $\wedge$ q)] is a tautology:

$\Rightarrow \neg(\neg p \vee q) \vee [\neg(r \wedge p) \vee (r \wedge q)]$

$\Rightarrow (p \wedge \neg q) \vee [(\neg r \vee \neg p) \vee (r \wedge q)]$

$\Rightarrow (p \wedge \neg q) \vee [((\neg r \vee \neg p) \vee r) \wedge ((\neg r \vee \neg p) \vee q)]$

$\Rightarrow (p \wedge \neg q) \vee [(r \vee \neg r \vee \neg p) \wedge (q \vee \neg r \vee \neg p)]$

$\Rightarrow (p \wedge \neg q) \vee [(q \vee \neg r \vee \neg p)]$

$\Rightarrow (q \vee \neg r \vee \neg p \vee p) \wedge (q \vee \neg r \vee \neg p \vee \neg q)$

$\Rightarrow T \wedge T$

$\Rightarrow T.$

Resolution **predicate logic**

Resolution is a theorem proving technique that proceeds by building refutation proofs, i.e., proofs by contradictions. It was invented by a Mathematician John Alan Robinson in the year 1965.

Resolution is used, if there are various statements are given, and we need to prove a conclusion of those statements. Unification is a key concept in proofs by resolutions. Resolution is a single inference rule which can efficiently operate on the **conjunctive normal form or clausal form**.

Clause: Disjunction of literals (an atomic sentence) is called a **clause**. It is also known as a unit clause.

Conjunctive Normal Form: A sentence represented as a conjunction of clauses is said to be **conjunctive normal form** or **CNF**.

The resolution inference rule:

The resolution rule for first-order logic is simply a lifted version of the propositional rule. Resolution can resolve two clauses if they contain complementary literals, which are assumed to be standardized apart so that they share no variables.

$$\frac{l_1 \vee \ldots\ldots \vee l_k, \qquad m_1 \vee \ldots\ldots \vee m_n}{\text{SUBST}(\theta, l_1 \vee \ldots\ldots \vee l_{i-1} \vee l_{i+1} \vee \ldots \vee l_k \vee m_1 \vee \ldots\ldots \vee m_{j-1} \vee m_{j+1} \vee \ldots\ldots \vee m_n)}$$

Where $\mathbf{l_i}$ and $\mathbf{m_j}$ are complementary literals.

This rule is also called the **binary resolution rule** because it only resolves exactly two literals.

Example:

We can resolve two clauses which are given below:

[Animal (g(x) V Loves (f(x), x)] and [¬ Loves(a, b) V ¬Kills(a, b)]

Where two complimentary literals are: **Loves (f(x), x) and ¬ Loves (a, b)**

These literals can be unified with unifier **θ= [a/f(x), and b/x]** , and it will generate a resolvent clause:

[Animal (g(x) V ¬ Kills(f(x), x)].

Steps for Resolution:

1. Conversion of facts into first-order logic.
2. Convert FOL statements into CNF
3. Negate the statement which needs to prove (proof by contradiction)
4. Draw resolution graph (unification).

To better understand all the above steps, we will take an example in which we will apply resolution.

Example:

a. **John likes all kind of food.**
b. **Apple and vegetable are food**
c. **Anything anyone eats and not killed is food.**
d. **Anil eats peanuts and still alive**

e. **Harry eats everything that Anil eats.**
   Prove by resolution that:

f. **John likes peanuts.**

Step-1: Conversion of Facts into FOL

In the first step we will convert all the given statements into its first order logic.

a. ∀x: food(x) → likes(John, x)
b. food(Apple) ∧ food(vegetables)
c. ∀x ∀y: eats(x, y) ∧ ¬ killed(x) → food(y)
d. eats (Anil, Peanuts) ∧ alive(Anil).
e. ∀x : eats(Anil, x) → eats(Harry, x)
f. ∀x: ¬ killed(x) → alive(x) } **added predicates.**
g. ∀x: alive(x) →¬ killed(x) }
h. likes(John, Peanuts)

Step-2: Conversion of FOL into CNF

In First order logic resolution, it is required to convert the FOL into CNF as CNF form makes easier for resolution proofs.

- **Eliminate all implication (→) and rewrite**
  a. ∀x ¬ food(x) V likes(John, x)
  b. food(Apple) Λ food(vegetables)
  c. ∀x ∀y ¬ [eats(x, y) Λ ¬ killed(x)] V food(y)
  d. eats (Anil, Peanuts) Λ alive(Anil)
  e. ∀x ¬ eats(Anil, x) V eats(Harry, x)
  f. ∀x¬ [¬ killed(x)] V alive(x)
  g. ∀x ¬ alive(x) V ¬ killed(x)
  h. likes(John, Peanuts).
- **Move negation (¬)inwards and rewrite**
  a. ∀x ¬ food(x) V likes(John, x)
  b. food(Apple) Λ food(vegetables)

c. $\forall x \forall y \neg$ eats(x, y) V killed(x) V food(y)

d. eats (Anil, Peanuts) Λ alive(Anil)

e. $\forall x \neg$ eats(Anil, x) V eats(Harry, x)

f. $\forall x \neg$killed(x)] V alive(x)

g. $\forall x \neg$ alive(x) V $\neg$ killed(x)

h. likes(John, Peanuts).

- **Rename variables or standardize variables**

  a. $\forall x \neg$ food(x) V likes(John, x)

  b. food(Apple) Λ food(vegetables)

  c. $\forall y \forall z \neg$ eats(y, z) V killed(y) V food(z)

  d. eats (Anil, Peanuts) Λ alive(Anil)

  e. $\forall w \neg$ eats(Anil, w) V eats(Harry, w)

  f. $\forall g \neg$killed(g)] V alive(g)

  g. $\forall k \neg$ alive(k) V $\neg$ killed(k)

  h. likes(John, Peanuts).

- **Eliminate existential instantiation quantifier by elimination.**
  In this step, we will eliminate existential quantifier $\exists$, and this process is known as **Skolemization**. But in this example problem since there is no existential quantifier so all the statements will remain same in this step.

- **Drop Universal quantifiers.**
  In this step we will drop all universal quantifier since all the statements are not implicitly quantified so we don't need it.

  a. $\neg$ food(x) V likes(John, x)

  b. food(Apple)

  c. food(vegetables)

  d. $\neg$ eats(y, z) V killed(y) V food(z)

  e. eats (Anil, Peanuts)

  f. alive(Anil)

  g. $\neg$ eats(Anil, w) V eats(Harry, w)

  h. killed(g) V alive(g)

i. ¬ alive(k) V ¬ killed(k)

j. likes(John, Peanuts).

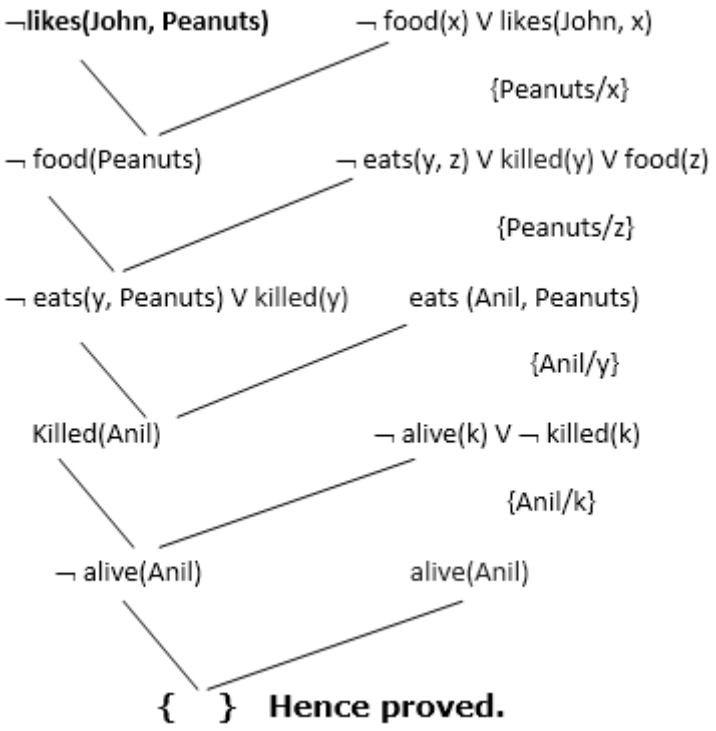
- **Distribute conjunction ∧ over disjunction ¬.**
  This step will not make any change in this problem.

Step-3: Negate the statement to be proved

In this statement, we will apply negation to the conclusion statements, which will be written as ¬likes(John, Peanuts)

Step-4: Draw Resolution graph:

Now in this step, we will solve the problem by resolution tree using substitution. For the above problem, it will be given as follows:



Hence the negation of the conclusion has been proved as a complete contradiction with the given set of statements.

Explanation of Resolution graph:

- In the first step of resolution graph, **¬likes(John, Peanuts)** , and **likes(John, x)** get resolved(canceled) by substitution of **{Peanuts/x}**, and we are left with ¬ **food(Peanuts)**

- In the second step of the resolution graph, ¬ **food(Peanuts)** , and **food(z)** get resolved (canceled) by substitution of **{ Peanuts/z}**, and we are left with ¬ **eats(y, Peanuts) V killed(y)** .
- In the third step of the resolution graph, ¬ **eats(y, Peanuts)** and **eats (Anil, Peanuts)** get resolved by substitution **{Anil/y}**, and we are left with **Killed(Anil)** .
- In the fourth step of the resolution graph, **Killed(Anil)** and ¬ **killed(k)** get resolve by substitution **{Anil/k}**, and we are left with ¬ **alive(Anil)** .
- In the last step of the resolution graph ¬ **alive(Anil)** and **alive(Anil)** get resolved.

---

11. Differentiate monotonic, non-monotonic and abductive reasoning using suitable example.

Abductive reasoning:

Abductive reasoning is a form of logical reasoning which starts with single or multiple observations then seeks to find the most likely explanation or conclusion for the observation.

Abductive reasoning is an extension of deductive reasoning, but in abductive reasoning, the premises do not guarantee the conclusion.

Example:

Implication: Cricket ground is wet if it is raining

Axiom: Cricket ground is wet.

Conclusion It is raining.

Monotonic Reasoning:

In monotonic reasoning, once the conclusion is taken, then it will remain the same even if we add some other information to existing information in our knowledge base. In monotonic reasoning, adding knowledge does not decrease the set of prepositions that can be derived.

To solve monotonic problems, we can derive the valid conclusion from the available facts only, and it will not be affected by new facts.

Monotonic reasoning is not useful for the real-time systems, as in real time, facts get changed, so we cannot use monotonic reasoning.

Monotonic reasoning is used in conventional reasoning systems, and a logic-based system is monotonic.

Any theorem proving is an example of monotonic reasoning.

Example:

- Earth revolves around the Sun.

It is a true fact, and it cannot be changed even if we add another sentence in knowledge base like, "The moon revolves around the earth" Or "Earth is not round," etc**.**

Advantages of Monotonic Reasoning:

- In monotonic reasoning, each old proof will always remain valid.
- If we deduce some facts from available facts, then it will remain valid for always.

Disadvantages of Monotonic Reasoning:

- We cannot represent the real world scenarios using Monotonic reasoning.
- Hypothesis knowledge cannot be expressed with monotonic reasoning, which means facts should be true.
- Since we can only derive conclusions from the old proofs, so new knowledge from the real world cannot be added.

Non-monotonic Reasoning

In Non-monotonic reasoning, some conclusions may be invalidated if we add some more information to our knowledge base.

Logic will be said as non-monotonic if some conclusions can be invalidated by adding more knowledge into our knowledge base.

Non-monotonic reasoning deals with incomplete and uncertain models.

"Human perceptions for various things in daily life, "is a general example of non-monotonic reasoning.

Example: Let suppose the knowledge base contains the following knowledge:

- Birds can fly
- Penguins cannot fly
- Pitty is a bird

So from the above sentences, we can conclude that Pitty can fly.

However, if we add one another sentence into knowledge base "Pitty is a penguin", which concludes "Pitty cannot fly", so it invalidates the above conclusion.

Advantages of Non-monotonic reasoning:

- **For real-world systems such as Robot navigation, we can use non-monotonic reasoning.**
- **In Non-monotonic reasoning, we can choose probabilistic facts or can make assumptions.**

Disadvantages of Non-monotonic Reasoning:

- **In non-monotonic reasoning, the old facts may be invalidated by adding new sentences.**
- **It cannot be used for theorem proving.**

Monotonic Reasoning vs Non-monotonic Reasoning

	Monotonic Reasoning	Non-Monotonic Reasoning
1	Monotonic Reasoning is the process which does not change its direction or can say that it moves in the one direction.	Non-monotonic Reasoning is the process which changes its direction or values as the knowledge base increases.
2	Monotonic Reasoning deals with very specific type of models, which has valid proofs.	Non-monotonic reasoning deals with incomplete or not known facts.
3	The addition in knowledge won't change the result.	The addition in knowledge will invalidate the previous conclusions and change the result.

4	**In monotonic reasoning, results are always true, therefore, set of prepositions will only increase.**	**In non-monotonic reasoning, results and set of prepositions will increase and decrease based on condition of added knowledge.**
5	**Monotonic Reasoning is based on true facts.**	**Non-monotonic Reasoning is based on assumptions.**
6	**Deductive Reasoning is the type of monotonic reasoning.**	**Abductive Reasoning and Human Reasoning is a non-monotonic type of reasoning.**

Definitions:

Monotonic means something that does not vary or change.

Non-Monotonic means something which can vary according to the situation or condition.

Monotonic Reasoning

Monotonic Reasoning is the process that does not change its direction or can say that it moves in the one direction.

- **Monotonic Reasoning will move in the same direction continuously means it will either move in increasing order or decrease.**
- **But since Monotonic Reasoning depends on knowledge and facts, It will only increase and will never decrease in this reasoning.**
- **Example:**
    - **Sun rises in the East and sets in the West.**

Non-monotonic Reasoning

Non-monotonic Reasoning is the process that changes its direction or values as the knowledge base increases.

- **It is also known as NMR in Artificial Intelligence.**
- **Non-monotonic Reasoning will increase or decrease based on the condition.**
- **Since that Non-monotonic Reasoning depends on assumptions, It will change itself with improving knowledge or facts.**
- **Example:**
    - **Consider a bowl of water, If we put it on the stove and turn the flame on it will obviously boil hot and as we will turn off the flame it will cool down gradually.**

---

12. Explain Bayes theorem. Write some applications of Bayes' theorem.

Bayes' theorem:

Bayes' theorem is also known as **Bayes' rule, Bayes' law**, or **Bayesian reasoning**, which determines the probability of an event with uncertain knowledge.

In probability theory, it relates the conditional probability and marginal probabilities of two random events.

Bayes' theorem was named after the British mathematician **Thomas Bayes**. The **Bayesian inference** is an application of Bayes' theorem, which is fundamental to Bayesian statistics.

It is a way to calculate the value of P(B|A) with the knowledge of P(A|B).

Bayes' theorem allows updating the probability prediction of an event by observing new information of the real world.

Example: If cancer corresponds to one's age then by using Bayes' theorem, we can determine the probability of cancer more accurately with the help of age.

Bayes' theorem can be derived using product rule and conditional probability of event A with known event B:

As from product rule we can write:

1. $P(A \wedge B) = P(A|B)\, P(B)$ or

Similarly, the probability of event B with known event A:

1. $P(A \wedge B) = P(B|A)\, P(A)$

Equating right hand side of both the equations, we will get:

$$P(A|B) = \frac{P(B|A)\, P(A)}{P(B)} \quad \text{....(a)}$$

The above equation (a) is called as **Bayes' rule** or **Bayes' theorem**. This equation is basic of most modern AI systems for **probabilistic inference**.

It shows the simple relationship between joint and conditional probabilities. Here,

P(A|B) is known as **posterior**, which we need to calculate, and it will be read as Probability of hypothesis A when we have occurred an evidence B.

P(B|A) is called the likelihood, in which we consider that hypothesis is true, then we calculate the probability of evidence.

P(A) is called the **prior probability**, probability of hypothesis before considering the evidence

P(B) is called **marginal probability**, pure probability of an evidence.

In the equation (a), in general, we can write P (B) = P(A)*P(B|Ai), hence the Bayes' rule can be written as:

$$P(A_i|B) = \frac{P(A_i) * P(B|A_i)}{\sum_{i=1}^{k} P(A_i) * P(B|A_i)}$$

Where A_1, A_2, A_3,....., A_n is a set of mutually exclusive and exhaustive events.

Applying Bayes' rule:

Bayes' rule allows us to compute the single term P(B|A) in terms of P(A|B), P(*B*), and P(A). This is very useful in cases where we have a good probability of these three terms and want to determine the fourth one. Suppose we want to perceive the effect of some unknown cause, and want to compute that cause, then the Bayes' rule becomes:

$$\mathbf{P(cause|effect)} = \frac{\mathrm{P(effect|cause)\ P(cause)}}{\mathrm{P(effect)}}$$

Application of Bayes' theorem in Artificial intelligence:

Following are some applications of Bayes' theorem:

- It is used to calculate the next step of the robot when the already executed step is given.
- Bayes' theorem is helpful in weather forecasting.
- It can solve the Monty Hall problem.

---

13. Write a note on forward and backward reasoning.

Reasoning

The reasoning is the mental process of deriving logical conclusion and making predictions from available knowledge, facts, and beliefs. Or we can say, "Reasoning is a way to infer facts from existing data." It is a general process of thinking rationally, to find valid conclusions.

forward reasoning

- a problem generally includes the initial data and facts in order to arrive at the solution.
- These unknown facts and information is used to deduce the result.
- For example, while diagnosing a patient the doctor first check the symptoms and medical condition of the body such as temperature, blood pressure, pulse, eye colour, blood, etcetera. After that, the patient symptoms are analysed and compared against the predetermined symptoms.
- Then the doctor is able to provide the medicines according to the symptoms of the patient. So, when a solution employs this manner of reasoning, it is known as forward rereasoning.

Forward reasoning steps

1. In the first step, the system is given one or more than one constraints.

2. Then the rules are searched in the knowledge base for each constraint.

3. The rules that fulfil the condition are selected(i.e., IF part).

4. Now each rule is able to produce new conditions from the conclusion of the invoked one.

5. As a result, THEN part is again included in the existing one.

6. The added conditions are processed again by repeating step 2.

7. The process will end if there is no new conditions exist.

backward reasoning

- The backward reasoning is inverse of forward reasoning in which goal is analysed in order to deduce the rules, initial facts and data.
- We can understand the concept by the similar example given in the above definition, where the doctor is trying to diagnose the patient with the help of the inceptive data such as symptoms.
- However, in this case, the patient is experiencing a problem in his body, on the basis of which the doctor is going to prove the symptoms.
- This kind of reasoning comes under backward reasoning.

backward reasoning steps

1. Firstly, the goal state and the rules are selected where the goal state reside in the THEN part as the conclusion.
2. From the IF part of the selected rule the subgoals are made to be satisfied for the goal state to be true.
3. Set initial conditions important to satisfy all the subgoals.
4. Verify whether the provided initial state matches with the established states.
5. If it fulfils the condition then the goal is the solution otherwise other goal state is selected.

---

14. Discuss Conceptual Dependency with the help of suitable example.

Conceptual Dependency

Conceptual Dependency originally developed to represent knowledge acquired from natural language input.

Conceptual Dependency originally developed to represent knowledge

acquired from natural language input.

The **goals** of this theory are:

- To help in the drawing of inference from sentences.
- To be independent of the words used in the original input.
- That is to say: For any 2 (or more) sentences that are identical in meaning there should be only one representation of that meaning.
- It has been used by many programs that portend to understand English
- (MARGIE, SAM, PAM).
- CD developed by Schank et al

CD provides:

1. a structure into which nodes representing information can be placed
2. a specific set of primitives
3. at a given level of granularity.

Sentences are represented as a series of diagrams depicting actions

1. using both abstract and real physical situations.
2. The agent and the objects are represented
3. The actions are built up from a set of primitive acts which can be modified by tense.

Primitive Acts are:

- ATRANS--Transfer of an abstract relationship. e.g. give.
- PTRANS--Transfer of the physical location of an object. e.g. go.
- PROPEL--Application of a physical force to an object. e.g. push.
- MTRANS--Transfer of mental information. e.g. tell.
- MBUILD--Construct new information from old. e.g. decide.
- SPEAK-- Utter a sound. e.g. say.
- ATTEND-- Focus a sense on a stimulus. e.g. listen, watch.

- MOVE-- Movement of a body part by owner. e.g. punch, kick.
- GRASP-- Actor grasping an object. e.g. clutch.
- INGEST--Actor ingesting an object. e.g. eat.
- EXPEL--Actor getting rid of an object from body. e.g. ????.

Six primitive conceptual categories provide building blocks which are the set of allowable dependencies in the concepts in a sentence:

1. PP-- Real world objects.
2. ACT-- Real world actions (An action done to an object).
3. PA--Attributes of objects.
4. AA-- Attributes of actions.
5. T–Times
6. LOC-- Locations.
7. Actor: The performer of an ACT.
8. Object: A thing that is acted upon.
9. Recipient: The reciever of an object as the result of an ACT.
10. Direction: The location that an ACT is directed toward.
11. State:The state that an object is in.

---

15. Explain scripts in detail using conceptual dependency with example (Script for going to the bank to withdraw money).

Scripts

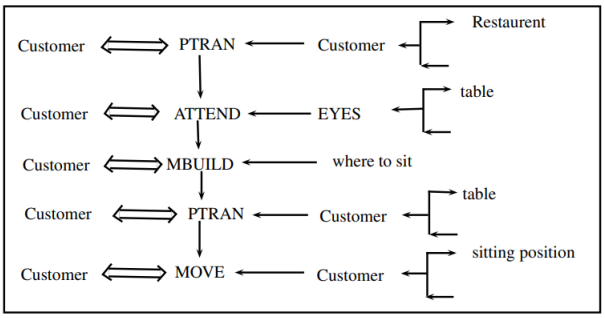
A script is a structure that prescribes a set of circumstances which could be expected to follow on from one another.

It is similar to a thought sequence or a chain of situations which could be anticipated.

It could be considered to consist of a number of slots or frames but with more specialised roles.

- The scripts represent stereotypical sequences of events, like going to a restaurant, buying from a store, etc.

- The theory of scripts has an emphasis to understand and quick access those events that always happen in a stereotypical event sequence, without worrying about those inferences which would most likely be irrelevant.
- Scripts are larger knowledge structures, used to solve problems using undirected inference.
- The scripts are pre-compiled sets of likely inferences, with elements of each set packed together so that they can be searched more efficiently, and produces lesser number of irrelevant inferences.
- A script is a set of roles (participants) involved in the script, as well as common objects used.
- A script comprises scenes, such that each scene describes typical events in a portion of a script.
- For example, in a restaurant's script, roles may be waiter, customer, restaurant itself, and food. Scene may be, ENTER, ORDER, EAT, PAY, LEAVE. Each scene's details are represented as a sequence of CD representations. For example, the ENTER scene in a restaurant may consist of a causal chain as shown in Fig. 7.18, and the sequence of CDs are:
    - Scene-name: ENTER
    - C PTRAN C into restaurant
    - C ATTEND eyes to tables
    - C MBUlLD where to sit
    - C PTRANS C to table
    - C MOVE C to sitting position
    - In above, C is a customer



Scripts are beneficial because:

Events tend to occur in known runs or patterns.

Causal relationships between events exist.

Entry conditions exist which allow an event to take place

Prerequisites exist upon events taking place.

E.g. when a student progresses through a degree scheme or when a purchaser buys a house.

The components of a script include:

Entry Conditions-- these must be satisfied before events in the script can occur.

Results-- Conditions that will be true after events in script occur.

Props-- Slots representing objects involved in events.

Roles-- Persons involved in the events.

Track--Variations on the script. Different tracks may share components of the same script.

Scenes--The sequence of events that occur. Events are represented in conceptual dependency form.

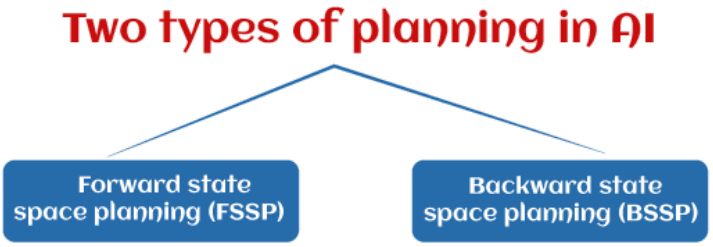
---

16. What is a Plan? Explain various types of planning and role of planning in artificial intelligence.

Plan:

We require domain description, task specification, and goal description for any planning system. A plan is considered a sequence of actions, and each action has its preconditions that must be satisfied before it can act and some effects that can be positive or negative.

So, we have **Forward State Space Planning (FSSP)** and **Backward State Space Planning (BSSP)** at the basic level.



1. Forward State Space Planning (FSSP)

FSSP behaves in the same way as forwarding state-space search. It says that given an initial state S in any domain, we perform some necessary actions and obtain a new state S' (which also contains some new terms), called a progression. It continues until we reach the target position. Action should be taken in this matter.

- **Disadvantage**: Large branching factor
- **Advantage**: The algorithm is Sound

2. Backward State Space Planning (BSSP)

BSSP behaves similarly to backward state-space search. In this, we move from the target state g to the sub-goal g, tracing the previous action to achieve that goal. This process is called regression (going back to the previous goal or sub-goal). These sub-goals should also be checked for consistency. The action should be relevant in this case.

- **Disadvantages**: not sound algorithm (sometimes inconsistency can be found)
- **Advantage**: Small branching factor (much smaller than FSSP)

So for an efficient planning system, we need to combine the features of FSSP and BSSP, which gives rise to target stack planning.

Target stack planning

It is one of the most important planning algorithms used by STRIPS.

Stacks are used in algorithms to capture the action and complete the target. A knowledge base is used to hold the current situation and actions.

A target stack is similar to a node in a search tree, where branches are created with a choice of action.

Role:

Planning is very important to make any AI project.

Planning is an important part of Artificial Intelligence which deals with the tasks and domains of a particular problem. Planning is considered the logical side of acting.

Everything we humans do is with a definite goal in mind, and all our actions are oriented towards achieving our goal. Similarly, Planning is also done for Artificial Intelligence.

For example, Planning is required to reach a particular destination. It is necessary to find the best route in Planning, but the tasks to be done at a particular time and why they are done are also very important.

That is why Planning is considered the logical side of acting. In other words, Planning is about deciding the tasks to be performed by the artificial intelligence system and the system's functioning under domain-independent conditions.

Planning in artificial intelligence is about decision-making actions performed by robots or computer programs to achieve a specific goal.

Execution of the plan is about choosing a sequence of tasks with a high probability of accomplishing a specific task.

Components of the planning system

The plan includes the following important steps:

- Choose the best rule to apply the next rule based on the best available guess.
- Apply the chosen rule to calculate the new problem condition.
- Find out when a solution has been found.
- Detect dead ends so they can be discarded and direct system effort in more useful directions.
- Find out when a near-perfect solution is found.

Non-linear Planning

This Planning is used to set a goal stack and is included in the search space of all possible sub-goal orderings. It handles the goal interactions by the interleaving method.

---

17. Explain Natural Language Processing. Why it is used?

Using a program or, indeed, a computer that can manipulate or comprehend speech through text is known as natural language processing (NLP). Comparison examples are human interaction, understanding one another's viewpoints, and responding properly. In NLP, computers can perform that communication, comprehension, and response instead of humans.

Features of NLP

1. Morphological Processing

NLP's initial element is morphology analysis. It involves splitting up large linguistic input blocks smaller groups of tokens that represent phrases, sections, as well as phrases. Any term like "daily," for instance, can indeed be split down into two sub-word tokens as "ever other."

2. Syntax analysis

One of the most crucial parts of NLP is the second element, syntax analysis. The following are indeed the goals of just this element:

- to determine whether a phrase is properly crafted.
- to organise it within a framework which demonstrates underlying grammatical connections between the different words.
- Examples include statements such "The student walks towards the classroom," that could be disallowed by something like a syntax analyzer.

3. Semantic analysis

The third component of NLP, semantics evaluation, is utilised to assess the biblical text meaning. It involves extrapolating the biblical text specific meaning, or determining what the dictionaries would claim is its interpretation. E.g. The semantics analysis will ignore phrases like "It was a heated dessert."

4. Pragmatic analysis

In NLP, pragmatic advice comes in at number four. It involves tying item connections discovered by the earlier element, or sentiment analysis, to the actual objects or events that occur within every scenario. E.g. Put the fruits in the basket on the table. Because this statement can now have two different semantic readings, the pragmatist analysis may select either of the following options.

5. Morphological Processing:

Besides splitting the input into smaller groups of tokens, morphological processing also involves identifying the base form of words, lemmatization, and the different inflected forms of words, known as stemming. These techniques help NLP systems understand the relationships between different forms of words and can improve the accuracy of downstream tasks such as sentiment analysis.

6. Syntax analysis:

Syntax analysis involves determining if a sentence is properly constructed and understanding the relationships between different parts of a sentence. This includes identifying subjects, objects, verbs, and other parts of speech, as well as understanding the different grammatical structures of a language. This knowledge is critical for tasks such as machine translation, where understanding the syntax of the source and target languages is essential.

7. Semantic analysis:

Semantic analysis involves extracting meaning from text and understanding the relationships between words and concepts. This includes identifying synonyms and antonyms, understanding word sense disambiguation, and recognizing the relationships between different entities in a sentence. These techniques are essential for tasks such as question-answering systems or chatbots that require a deep understanding of natural language.

8. Pragmatic analysis:

The pragmatic analysis involves understanding the context in which language is used and identifying the intended meaning behind a sentence. This includes understanding sarcasm, irony, or humor and recognizing when a sentence has multiple interpretations. The pragmatic analysis is particularly important for applications such as sentiment analysis, where understanding a text's underlying tone and context can greatly improve the accuracy of the analysis.

What is natural language processing used for?

Some of the main functions that natural language processing algorithms perform are:

- **Text classification.** This involves assigning tags to texts to put them in categories. This can be useful for sentiment analysis, which helps the natural language processing algorithm determine the sentiment, or emotion behind a text. For example, when brand A is mentioned in X number of texts, the algorithm can determine how many of those mentions were positive and how many were negative. It can also be useful for intent detection, which helps predict what the speaker or writer may do based on the text they are producing.
- **Text extraction.** This involves automatically summarizing text and finding important pieces of data. One example of this is keyword extraction, which pulls the most important words from the text, which can be useful for search engine optimization. Doing this with natural language processing requires some programming -- it is not completely automated. However, there are plenty of simple keyword extraction tools that automate most of the process -- the user just has to set parameters within the program. For example, a tool might pull out the most frequently used words in the text. Another example is named entity recognition, which extracts the names of people, places and other entities from text.
- **Machine translation.** This is the process by which a computer translates text from one language, such as English, to another language, such as French, without human intervention.
- **Natural language generation.** This involves using natural language processing algorithms to analyze unstructured data and automatically produce content based on that data. One example of this is in language models such as GPT3, which are able to analyze an unstructured text and then generate believable articles based on the text.

The functions listed above are used in a variety of real-world applications, including:

- customer feedback analysis -- where AI analyzes social media reviews;

- customer service automation -- where voice assistants on the other end of a customer service phone line are able to use speech recognition to understand what the customer is saying, so that it can direct the call correctly;
- automatic translation -- using tools such as Google Translate, Bing Translator and Translate Me;
- academic research and analysis -- where AI is able to analyze huge amounts of academic material and research papers not just based on the metadata of the text, but the text itself;
- analysis and categorization of medical records -- where AI uses insights to predict, and ideally prevent, disease;
- word processors used for plagiarism and proofreading -- using tools such as Grammarly and Microsoft Word;
- stock forecasting and insights into financial trading -- using AI to analyze market history and 10-K documents, which contain comprehensive summaries about a company's financial performance;
- talent recruitment in human resources; and
- automation of routine litigation tasks -- one example is the artificially intelligent attorney.

---

18. Describe Mini-Max algorithm.

Mini-Max Algorithm in Artificial Intelligence

- Mini-max algorithm is a recursive or backtracking algorithm which is used in decision-making and game theory.
- It provides an optimal move for the player assuming that opponent is also playing optimally.
- Mini-Max algorithm uses recursion to search through the game-tree.
- Min-Max algorithm is mostly used for game playing in AI. Such as Chess, Checkers, tic-tac-toe, go, and various tow-players game.

- This Algorithm computes the minimax decision for the current state.
- In this algorithm two players play the game, one is called MAX and other is called MIN.
- Both the players fight it as the opponent player gets the minimum benefit while they get the maximum benefit.
- Both Players of the game are opponent of each other, where MAX will select the maximized value and MIN will select the minimized value.
- The minimax algorithm performs a depth-first search algorithm for the exploration of the complete game tree.
- The minimax algorithm proceeds all the way down to the terminal node of the tree, then backtrack the tree as the recursion.

Initial call:

Minimax(node, 3, true)

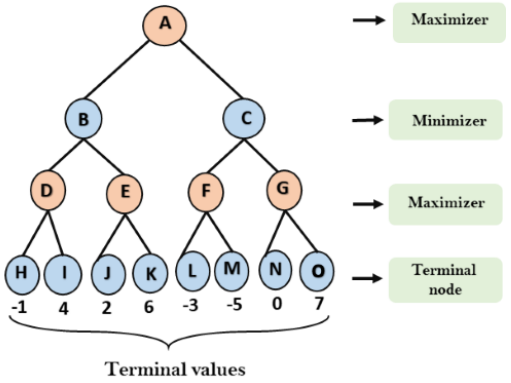
Working of Min-Max Algorithm:

- The working of the minimax algorithm can be easily described using an example. Below we have taken an example of game-tree which is representing the two-player game.
- In this example, there are two players one is called Maximizer and other is called Minimizer.
- Maximizer will try to get the Maximum possible score, and Minimizer will try to get the minimum possible score.
- This algorithm applies DFS, so in this game-tree, we have to go all the way through the leaves to reach the terminal nodes.
- At the terminal node, the terminal values are given so we will compare those value and backtrack the tree until the initial state occurs.

Following are the main steps involved in solving the two-player game tree:

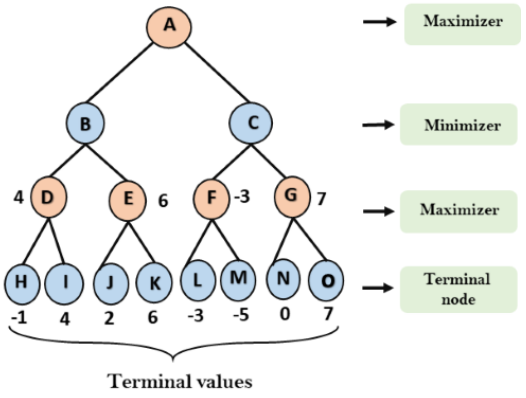
Step-1: In the first step, the algorithm generates the entire game-tree and apply the utility function to get the utility values for the terminal states. In the below tree diagram, let's take A is the initial state of the tree. Suppose maximizer takes first turn which has worst-case

initial value =- infinity, and minimizer will take next turn which has worst-case initial value = +infinity.



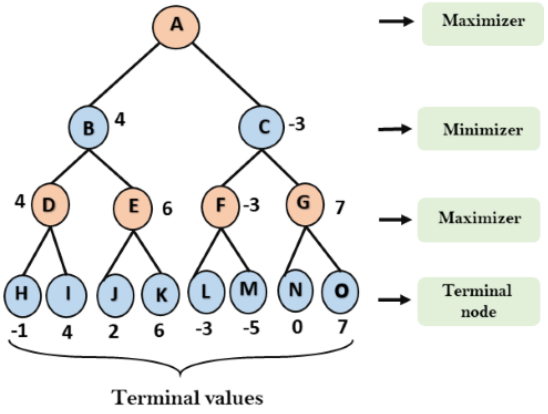
Step 2: Now, first we find the utilities value for the Maximizer, its initial value is -∞, so we will compare each value in terminal state with initial value of Maximizer and determines the higher nodes values. It will find the maximum among the all.

- For node D max(-1,- -∞) => max(-1,4)= 4
- For Node E max(2, -∞) => max(2, 6)= 6
- For Node F max(-3, -∞) => max(-3,-5) = -3
- For node G max(0, -∞) = max(0, 7) = 7



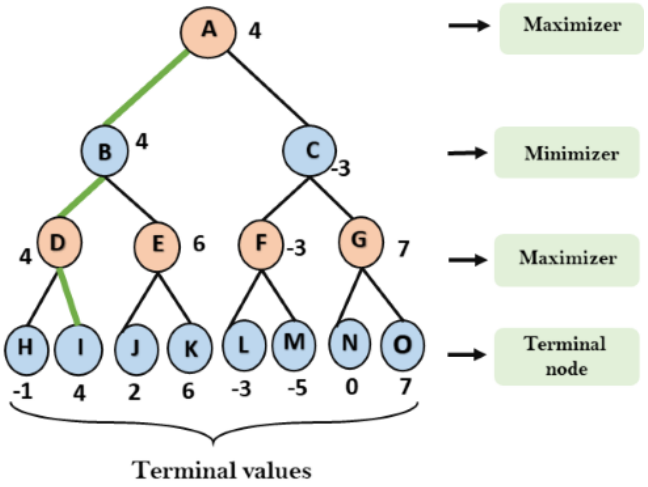
Step 3: In the next step, it's a turn for minimizer, so it will compare all nodes value with +∞, and will find the 3rd layer node values.

- For node B= min(4,6) = 4
- For node C= min (-3, 7) = -3



Step 4: Now it's a turn for Maximizer, and it will again choose the maximum of all nodes value and find the maximum value for the root node. In this game tree, there are only 4 layers, hence we reach immediately to the root node, but in real games, there will be more than 4 layers.

- For node A max(4, -3)= 4



That was the complete workflow of the minimax two player game.

Properties of Mini-Max algorithm:

- **Complete-** Min-Max algorithm is Complete. It will definitely find a solution (if exist), in the finite search tree.
- **Optimal-** Min-Max algorithm is optimal if both opponents are playing optimally.

- **Time complexity-** As it performs DFS for the game-tree, so the time complexity of Min-Max algorithm is **$O(b^m)$**, where b is branching factor of the game-tree, and m is the maximum depth of the tree.
- **Space Complexity-** Space complexity of Mini-max algorithm is also similar to DFS which is **O(bm)**.

Limitation of the minimax Algorithm:

The main drawback of the minimax algorithm is that it gets really slow for complex games such as Chess, go, etc. This type of games has a huge branching factor, and the player has lots of choices to decide. This limitation of the minimax algorithm can be improved from **alpha-beta pruning** which we have discussed in the next topic.

---

19. Discuss neural networks.
What are the advantages of neural networks?

- A neural network is a method in artificial intelligence that teaches computers to process data in a way that is inspired by the human brain.
- It is a type of machine learning process, called deep learning, that uses interconnected nodes or neurons in a layered structure that resembles the human brain.
- It creates an adaptive system that computers use to learn from their mistakes and improve continuously.
- Thus, artificial neural networks attempt to solve complicated problems, like summarizing documents or recognizing faces, with greater accuracy.

How do neural networks work?

- The human brain is the inspiration behind neural network architecture. Human brain cells, called neurons, form a complex, highly interconnected network and send electrical signals to each other to help humans process information.
- Similarly, an artificial neural network is made of artificial neurons that work together to solve a problem. Artificial neurons are software modules, called nodes, and artificial neural networks are software programs or algorithms that, at their core, use computing systems to solve mathematical calculations.

Simple neural network architecture

A basic neural network has interconnected artificial neurons in three layers:

Input Layer

Information from the outside world enters the artificial neural network from the input layer. Input nodes process the data, analyze or categorize it, and pass it on to the next layer.

Hidden Layer

Hidden layers take their input from the input layer or other hidden layers. Artificial neural networks can have a large number of hidden layers. Each hidden layer analyzes the output from the previous layer, processes it further, and passes it on to the next layer.

Output Layer

The output layer gives the final result of all the data processing by the artificial neural network. It can have single or multiple nodes. For instance, if we have a binary (yes/no) classification problem, the output layer will have one output node, which will give the result as 1 or 0. However, if we have a multi-class classification problem, the output layer might consist of more than one output node.

What are the **types of neural networks?**

Artificial neural networks can be categorized by how the data flows from the input node to the output node. Below are some examples:

Feedforward neural networks

- Feedforward neural networks process data in one direction, from the input node to the output node.
- Every node in one layer is connected to every node in the next layer.
- A feedforward network uses a feedback process to improve predictions over time.

Backpropagation algorithm

- Artificial neural networks learn continuously by using corrective feedback loops to improve their predictive analytics.
- In simple terms, you can think of the data flowing from the input node to the output node through many different paths in the neural network.
- Only one path is the correct one that maps the input node to the correct output node.
- To find this path, the neural network uses a feedback loop.

Convolutional neural networks

- The hidden layers in convolutional neural networks perform specific mathematical functions, like summarizing or filtering, called convolutions.
- They are very useful for image classification because they can extract relevant features from images that are useful for image recognition and classification.
- The new form is easier to process without losing features that are critical for making a good prediction.
- Each hidden layer extracts and processes different image features, like edges, color, and depth.

What Are The **Advantages of Neural Networks**

There are various advantages of neural networks, some of which are discussed below:

1) **Store information on the entire network**

Just like it happens in traditional programming where information is stored on the network and not on a database. If a few pieces of information disappear from one place, it does not stop the whole network from functioning.

2) **The ability to work with insufficient knowledge:**

After the training of ANN, the output produced by the data can be incomplete or insufficient. The importance of that missing information determines the lack of performance.

3) **Good falt tolerance**:

The output generation is not affected by the corruption of one or more than one cell of artificial neural network. This makes the networks better at tolerating faults.

4) **Distributed memory:**

For an artificial neural network to become able to learn, it is necessary to outline the examples and to teach it according to the output that is desired by showing those examples to the network. The progress of the network is directly proportional to the instances that are selected.

5) **Gradual Corruption**:

Indeed a network experiences relative degradation and slows over time. But it does not immediately corrode the network.

6) **Ability to train machine:**

ANN learn from events and make decisions through commenting on similar events.

7) **The ability of parallel processing:**

These networks have numerical strength which makes them capable of performing more than one function at a time.

---

20. Explain an expert system with example.
What are the characteristics of an expert system?
List the advantages of an expert system.

- An expert system is a computer program that is designed to solve complex problems and to provide decision-making ability like a human expert.
- It performs this by extracting knowledge from its knowledge base using the reasoning and inference rules according to the user queries.
- The expert system is a part of AI, and the first ES was developed in the year 1970, which was the first successful approach of artificial intelligence.
- It solves the most complex issue as an expert by extracting the knowledge stored in its knowledge base.
- The system helps in decision making for complex problems using **both facts and heuristics like a human expert**.
- It is called so because it contains the expert knowledge of a specific domain and can solve any complex problem of that particular domain.
- These systems are designed for a specific domain, such as **medicine, science,** etc.
- The performance of an expert system is based on the expert's knowledge stored in its knowledge base.
- The more knowledge stored in the KB, the more that system improves its performance.
- One of the common examples of an ES is a suggestion of spelling errors while typing in the Google search box.

Here, we will explain the working of an expert system by taking an example of MYCIN ES. Below are some steps to build an MYCIN:

- Firstly, ES should be fed with expert knowledge. In the case of MYCIN, human experts specialized in the medical field of bacterial infection, provide information about the causes, symptoms, and other knowledge in that domain.
- The KB of the MYCIN is updated successfully. In order to test it, the doctor provides a new problem to it. The problem is to identify the presence of the bacteria by inputting the details of a patient, including the symptoms, current condition, and medical history.
- The ES will need a questionnaire to be filled by the patient to know the general information about the patient, such as gender, age, etc.
- Now the system has collected all the information, so it will find the solution for the problem by applying if-then rules using the inference engine and using the facts stored within the KB.
- In the end, it will provide a response to the patient by using the user interface.

Characteristics of Expert System

- **High Performance:** The expert system provides high performance for solving any type of complex problem of a specific domain with high efficiency and accuracy.
- **Understandable:** It responds in a way that can be easily understandable by the user. It can take input in human language and provides the output in the same way.
- **Reliable:** It is much reliable for generating an efficient and accurate output.
- **Highly responsive:** ES provides the result for any complex query within a very short period of time.

Components of Expert System

An expert system mainly consists of three components:

1. User Interface

It **is an interface that helps a non-expert user to communicate with the expert system to find a solution**.

2. Inference Engine(Rules of Engine)

- The inference engine is known as the brain of the expert system There are two types of inference engine:
- **Deterministic Inference engine**
- **Probabilistic Inference engine**

3. Knowledge Base

- The knowledgebase is a type of storage that stores knowledge acquired from the different experts of the particular domain. It is considered as big storage of knowledge. The more the knowledge base, the more precise will be the Expert System.

Components of Knowledge Base

- **Factual Knowledge**.
- **Heuristic Knowledge**
- **Knowledge Representation**
- **Knowledge Acquisitions**

Why Expert System?

Before using any technology, we must have an idea about why to use that technology and hence the same for the ES. Although we have human experts in every field, then what is the need to develop a computer-based system. So below are the points that are describing the need of the ES:

1. **No memory Limitations:** It can store as much data as required and can memorize it at the time of its application. But for human experts, there are some limitations to memorize all things at every time.
2. **High Efficiency:** If the knowledge base is updated with the correct knowledge, then it provides a highly efficient output, which may not be possible for a human.
3. **Expertise in a domain:** There are lots of human experts in each domain, and they all have different skills, different experiences, and different skills, so it is not easy to get a final output for the query. But if we put the knowledge gained from human

experts into the expert system, then it provides an efficient output by mixing all the facts and knowledge

4. **Not affected by emotions:** These systems are not affected by human emotions such as fatigue, anger, depression, anxiety, etc.. Hence the performance remains constant.
5. **High security:** These systems provide high security to resolve any query.
6. **Considers all the facts:** To respond to any query, it checks and considers all the available facts and provides the result accordingly. But it is possible that a human expert may not consider some facts due to any reason.
7. **Regular updates improve the performance:** If there is an issue in the result provided by the expert systems, we can improve the performance of the system by updating the knowledge base.
8. **Problem-solving:** It has problem-solving capabilities.
9. **Advising:** It is capable of advising the human being for the query of any domain from the particular ES.
10. **Provide decision-making capabilities:** It provides the capability of decision making in any domain, such as for making any financial decision, decisions in medical science, etc.

Advantages of Expert System

- These systems are highly reproducible.
- They can be used for risky places where the human presence is not safe.
- Error possibilities are less if the KB contains correct knowledge.
- The performance of these systems remains steady as it is not affected by emotions, tension, or fatigue.
- They provide a very high speed to respond to a particular query.

Limitations of Expert System

- The response of the expert system may get wrong if the knowledge base contains the wrong information.
- Like a human being, it cannot produce a creative output for different scenarios.

- Its maintenance and development costs are very high.
- Knowledge acquisition for designing is much difficult.
- For each domain, we require a specific ES, which is one of the big limitations.
- It cannot learn from itself and hence requires manual updates.

Applications of Expert System

- **In designing and manufacturing domain**
  It can be broadly used for designing and manufacturing physical devices such as camera lenses and automobiles.
- **In the knowledge domain**
  These systems are primarily used for publishing the relevant knowledge to the users. The two popular ES used for this domain is an advisor and a tax advisor.
- **In the finance domain**
  In the finance industries, it is used to detect any type of possible fraud, suspicious activity, and advise bankers that if they should provide loans for business or not.
- **In the diagnosis and troubleshooting of devices**
  In medical diagnosis, the ES system is used, and it was the first area where these systems were used.
- **Planning and Scheduling**
  The expert systems can also be used for planning and scheduling some particular tasks for achieving the goal of that task.