Saroj Dahal - 191725

Assesment exam.

Date
Page

1. a) "While humans are capable of exhibiting irrational behavior, it is imperative for Artifical Intelligence systems to adhere to rational decision-making process". Justify this statement. Also mention the requirements for a machine to pass the Turing Test.

Ans. Firstly, AI systems are designed to perform specific tasks with maximum efficiency and accuracy, and irrational behavior can compromise these goals. AI systems rely on algorithms and data to make decisions, and any irrational input can negatively affect their outputs. This is because Irrationality often arises from emotions, biases, or faulty assumptions, which can lead to errors in judgment and decision-making.

Secondly, AI systems are often used in critical applications such as healthcare, finance, and transportation, where even small errors can have serious consequences. In these contexts, it is crucial that AI systems make rational decisions based on objective data and logical reasoning. for example, in healthcare, an irrational decision by an AI system can lead to misdiagnosis, incorrect treatment, or even fatal outcomes.

Lastly, AI systems are not capable of exhibiting human emotions, which can cloud human judgment and lead to irrational decisions. In contrast, AI systems can analyze large amounts of data, recognize patterns, and make rational decisions based on objective criteria. This is especially useful in situations where emotions or personal biases can lead to suboptimal outcomes.

In conclusion, while humans may exhibit irrational behavior, it is imperative for AI systems to adhere to rational decision-making processes to ensure maximum efficiency, accuracy, and reliability in critical applications.

Second part:

The Turing test is a measure of a machine's ability to exhibit intillegent behavior that is indistinguishable from that of human being. The requirements for a machine to pass the Turing test are as follows:

- Natural language processing: The machine must be capable of understanding and processing human language, including spoken and written communication.

- Knowledge representation: The machine must be able to store and retrieve information, as well as make connections between pieces of information.

- Automated reasoning: The machine must be able to use information it has gathered to draw logical conclusions and make decisions.

- Machine learning: The machine must be able to learn from experience and adapt its behavior accordingly.

- Common sense reasoning: The machine must be able to make judgments based on contextual information and common sense.

- Creativity: The machine must be able to generate new and innovative ideas and solutions.

- Emotion: The machine must be able to understand and express emotions in a way that is indistinguishable from that of human being.

To pass the Turing test, a machine must be able to convince a

Date
Page

human evaluator that it is a human being by exhibiting all of these abilities in a way that is indistinguishable from a human being. It is important to note that passing the Turing test does not necessarily mean that a machine is truly intilligent as there may be ways to deceive the evaluator without actually exhibiting genuine intilligence. Nonetheless, passing the Turing test is still considered to be a significant milestone in the development of artifical intilligence.

b> Define the term Problem and Goal from an AI perspective. Discuss the importance of Means-end Analysis.

Ans. In the context of Artifical Intilligence (AI), a problem is defined as a situation where an agent needs to take some action to achieve a specific goal or set of goals, but the means to achieve the goal are not immediately obvious. In other words, a problem in AI refers to a situation where an agent needs to find a solution to a specific challenge or difficulty.

On the other hand, a goal is the objective or outcome that an agent seeks to achieve through its actions. Goals are essential in AI because they define the purpose or the reason for the agent's behavior, and they provide a measure of success for the agent's actions.

Mean-end Analysis:

Mean-end analysis is an important problem-solving startegy that is widely used in many fields, including Artifical Intelligence (AI), engineering, management, and decision-making. The primary importance of means-end analysis lies in its ability to break down complex problems into smaller, more manageable sub-problems, making them easier to solve.

Here are some of the key reasons why means-end analysis is important in problem-solving:

a. Helps in complex problem-solving: Complex problems can be overwhelming and difficult to solve, especially when they involve many variables and unknown factors. Mean-end analysis provides a framework for breaking down these complex problems into smaller, more manageable sub-problems that can be solved step by step.

b. Increase efficiency: By breaking down complex problems into smaller sub-problems, means-end analysis can help to organize the use of resources and increase efficiency. It allows agents to focus their efforts on the most critical issues and avoid wasting time and resources on less important factors.

c. Enhance decision-making: Means-end analysis can be used to evaluate different solutions and options for solving a problem. By breaking down the problem into smaller sub-problems and comparing different options, agents can make better-informed decision and choose the most effective solution.

d. Enables planning and goal-setting: Mean-end analysis can help agents to define their goals and develop a plan to achieve them. By identifying the differences between the current state and the goal state, agents can work out a series of steps to bridge that gap and achieve their objectives.

e. Provides a framework for learning: Means-end analysis can be used as a tool for learning and improvement. By analyzing the steps taken to solve a problem, agents can identify areas for improvement and adjust their approach to similar problems in the future.

Overall, means-end analysis is an important problem-solving strategy that provides a framework for breaking down complex problems

Date
Page

and achieving well-defined goals. It is a powerful tool that can help agents to optimize their resources, make informed decisions, and improve their problem-solving skills over time.

Q.N. 2.

a) "If you send me a e-mail message then I will finish writing the program", "If you do not send me an e-mail message then I will go to sleep early", and "If I go to sleep early then I will wake up feeling refreshed". Lead to conclusion "If I do not finish writing the program then I will wake up feeling refreshed". Prove by using propositional logic.

Ans. Let's assign the following propositions:

p : You send me a email message.

q : I finish writing the program.

r : I go to sleep early.

s : I wake up feeling refreshed.

Then we can translate the three given statements into propositional logic: (Hypothesis)

1. $p \rightarrow q$ Conclusion: $\sim q \rightarrow s$.
2. $\sim p \rightarrow r$
3. $r \rightarrow s$

S.N.	Steps	Reasons
1.	$p \rightarrow q$	given hypothesis (1)
2.	$\sim q \rightarrow \sim p$	using contrapositive (1)
3.	$\sim p \rightarrow r$	given hypothesis (2)
4.	$\sim q \rightarrow r$	using modus ponens (2) and (3)
5.	$r \rightarrow s$	given hypothesis (3)
6.	$\sim q \rightarrow s$	using modus ponens on (4) and (5).

b) Construct a frame with the following descriptions :
Mitthu is a cat. Jerry is a rat. Pukke is a penguin. A penguin is a bird. Cats and rats are mammals. Both birds and mammals are animals. Birds have legs and flies. Mammals have limbs and legs. Some animals fly and some do not. Mitthu and pukke are friends.

Ans.
Animal :

- Cat : Mitthu
  - Mammal.
- Rat : Jerry
  - Mammal.
- Penguin : Pukke.
  - Bird
  - Flies (some birds).
  - Has legs

Animal Categories :

- Mammal
  - Has limbs.
  - Has Legs.
- Bird.
  - Has legs
  - Flies

Friendship :

- Mitthu and Pukke are friends.

Date
Page

Q.N.3.

Q. How is the resolution algorithm used as a rule of inference in predicate logic? "All oversmart people are stupid. Children of all stupid persons are naughty. Roney is a children of Harry. Harry is over smart". Prove that "Roney is naughty" using a resolution algorithm.

ANS. The resolution algorithm is a rule of inference in predicate logic that is used to determine the validity of a statement by negating it and attempting to derive a contradiction from the resulting clauses. Here's how we can use the resolution algorithm to prove that "Roney is naughty" based on given statements:

i. Translate the given statements into predicate logic using variables and predicates:

$\forall x\,(\text{oversmart}(x) \rightarrow \text{stupid}(x))$

$\forall x\,\forall y\,(\text{stupid}(x) \wedge \text{parent}(y, x) \rightarrow \text{naughty}(y))$

parent (Harry, Roney)

oversmart (Harry).

ii. Negate the statement we want to prove (i.e. ~naughty (Roney)) and convert it into clause form:

naughty (x)

~naughty (Roney).

iii. Apply resolution to the clauses by resolving on the negated predicate and the second statement, which gives us the new clause: ~naughty (Harry):

$\sim$naughty (Roney), $\forall x\,\forall y\,(\text{stupid}(x) \wedge \text{parent}(y, x) \rightarrow \text{naughty}(y))$.

$\sim$naughty (Harry), $\forall x\,(\text{oversmart}(x) \rightarrow \text{stupid}(x))$

Date
Page

9. Apply resolution again to the new clause and the first statement, which gives us the empty clause (i.e., a contradiction):

$\sim$naughty (Harry), $\forall x$ (oversmart $(x) \rightarrow$ stupid(x))

stupid (Harry) (resolved from (ii) and (iii))

oversmart (Harry) $\rightarrow$ stupid (Harry) (resolved from (i) and (iii))

$\sim$oversmart (Harry) (derived by modus tollens by (iv)).

naughty (Roney) (resolved from (i) and (iv))

$\sim$naughty (Roney), naughty (Roney) (resolved from ii and v).

Since we have derived a contradiction, we can conclude that our initial assumption ($\sim$naughty (Roney)) must be false, and therefore "Roney is naughty" is true.

b) Represent the following statements into predicate logic:

i. Subik and Shrabin are players.

ii. All gardeners love the sun.

iii. Binay is not married.

iv. Swastima is married to Nischal.

v. Ram is Laxman's brother.

vi. You can fool all of the people some of the time.

vii. You can fool some of the people all of the time.

Ans.

i. Let $P(x)$ be "x is player". Then the statement "Subik and Shrabin are players" can be represented as:

$P(\text{Subik}) \wedge P(\text{Shrabin})$.

ii. Let $G(x)$ be "x is a gardener" and $S(x)$ be "x loves the sun". Then the statement "All gardeners love the sun" can be represented as:

$\forall x\ (G(x) \rightarrow S(x))$.

iii. Let M(x) be "x is married". Then the statement "Binay is not married" can be represented as:

$\sim M(Binay)$.

iv. Let M(x, y) be "x is married to y". Then the statement "Swastima is married to Nischal" can be represented as:

M(Swastima, Nischal).

v. Let B(x,y) be "x is the brother of y". Then the statement "Ram is Laxman's brother" can be represented as:

B(Ram, Laxman).

vi. Let F(x, y) be "x can fool y" and T(y) be "y is a person". Then the statement "You can fool all of the people some of the time" can be represented as:

$$\forall y\,(T(y) \rightarrow \exists x\, F(x,y))$$

vii. Let F(x,y) be "x can fool y" and T(y) be "y is a person". Then the statement "You can fool some of the people all of the time" can be represented as:

$$\exists x\, \forall y\,(T(y) \rightarrow F(x,y)).$$

Q.N.4.

a. Using genetic algorithm optimize the eq^n $f(x) = x^2$. [Take the sample size from 0 to 31 and use population size 4].

→ To optime, the eq^n $f(x) = x^2$ using a genetic algorithm w a population size of 4, we need to follow these steps:

a. Initialization: Generate an initial population of 4 random values of x within the range [0,31].

b. Fitness evaluation: Evaluate the fitness of each individual in the population by calculating its corresponding $f(x)$ value.

c. Selection: select two individuals from the populations for mating based on their fitness. We can use tournament selection or roulette wheel selection for this.

d. Crossover: Perform a crossover operation on the selected individuals to create two new offspring. We can use single-point crossover or two-point crossover for this.

e. Mutation: Apply mutation to the offspring generated in step (d) to introduce some random variation into the population. We can use the Gaussian mutation or uniform mutation for this.

f. Replacement: Replace the two least fit individuals in the population with the two offspring generated in step (d).

g. Repeat steps (b) to (f) for a fixed number of generations or until the optimal solution is found.

Date
Page

b) What do you mean by agent in AI? Differentiate between goal based agent and model based agent.

Ans. In AI, an agent is an autonomous entity that interacts with its environment to achieve certain goals. Agents can be designed to perform specific tasks, make decisions, or learn from their experiences. They are typically equipped with sensors to perceive the environment and effectors to act on the environment.

Agents can be designed to operate in various environments, including virtual and physical environments. Some examples of agents in AI include intelligent personal assistants like Siri and Alexa, self-driving cars, industrial robots, and game-playing AI agents.

The design of agents in AI can involve various approaches, including rule-based systems, decision trees, machine learning algorithms, and neural networks. The choice of approach depend on the complexity of the task and the nature of environment in which the agent is expected to operate.

Overall, the concept of an agent in AI provides a useful framework for designing and developing autonomous systems that can operate effectively in various environments and achieve specific goals.

Goal-based agents and model-based agents are two common types of agents in AI. The differences are as follows:

Goal-based agents	Model-based agents.
• They are designed to achieve specific pre-defined goals. They operate by considering their current state, the available actions they can take, and the potential outcomes of those actions to determine	• They use their understanding of the environment to build a model of the world, which the use to reason about the effects of their actions. They can oper even in complex and uncertain

Date
Page

	which action to take that best satisfies their goals.	environments by updating their model based on new observations, and adjusting their actions accordingly.
Focus :	Their focus is on achieving their pre-defined goals.	Their focus is on understanding and modeling the environment to optimize their actions.
Adaptability :	They are less adaptable than model-based agents because they operate based on pre-defined goals. They may not be able to adapt to new situations or unexpected events as effectively as model-based agents.	They are more adaptable because they can update their model of the world based on new observations.
Complexity :	Goal-based agents are simpler to design and implement than model-based agents because they do not require a model of the environment	Model-based agents are more complex to design and implement because they require a model of the environment.

Overall, while goal-based agents focus on achieving pre-defined goals, model-based agent focus on modeling the environment and adapting their actions based on their understanding of the environment. The choice between the two types of agents depends on the specific requirements of the task and the nature of the environment in which the agent will operate.

Date
Page

Q.N.5.

A. What are the similarities and differences between ANN and BNN? Discuss the reasons behind the popularity of ANN with its representation.

Ans. Artificial Neural Networks (ANN) and Bayesian Neural Networks (BNN) are two types of neural networks used in machine learning. While they share some similarities, they also have several key differences.

Similarities:

- Both ANN and BNN are based on the idea of connecting individual neurons or nodes to form a network.
- Both can learn to recognize patterns and relationships in data through a process called training.
- Both can be used for tasks such as classification, prediction, and decision making.

Differences:

- Both neural networks incorporate probabilistic modeling, which allows them to make predictions with uncertanity estimates. ANN do not incorporate probabilistic modeling.
- BNN use a Bayesian approach to learning and inference, while ANNs use a non-probabilistic approach.
- BNNs can provide better uncertanity estimates and can be more robust to overfitting, while ANNs can be more computationally efficient.

The popularity of ANN with its representation can be attributed to several reasons:

- ANN can be applied to a wide range of problems, from image recognition to natural language processing.
- ANN have a simple and intuitive representation, making them easy to understand and implement.

Date
Page

- ANN can learn from large amounts of data and generalize to new, unseen examples.
- ANN can be trained using backpropagation, a powerful optimization algorithm that is well-suited for deep learning.
- ANN can be used to build deep neural networks, which have demonstrated state-of-the-art performance on many benchmark tasks.

Overall, while both ANN and BNN share some similarities, they also have a several key differences. ANN's popularity can be attributed to their versatility, simplicity, and ability to learn from large amounts of data, as well as the powerful optimization techniques used for training them.

B. How is Neural Network trained? Discuss the perception learning and delta learning algorithm.

AN. A neural network is trained through a process called supervised learning, where the network is presented with a set of inputs along with their corresponding target outputs, and the network learns to map the inputs to the correct outputs through a series of weight adjustments.

The training process involves the following steps:

i. Initialization: The network is initialized with random weights.

ii. Forward propagation: The input data is fed forward through the network, and the outputs are computed. The difference between the predicted outputs and the target outputs ~~are~~ is calculated, and this difference is called the error.

iii. Backward propagation: The error is propagated backward through the network, and the weights are adjusted based on the contribution of each neuron to the error. This process is repeated for each training example.

iv. Update weights: The weights are adjusted to minimize the error. This is done using an optimization algorithm, such as stochastic gradient descent or Adam.

v. Repeat: Steps 2-4 are repeated for a fixed number of iterations or until the network reaches a satisfactory level of accuracy.

The process of training a neural network can be complex, and there are many variations of the training process depending on the type of network being used and the problem being solved.

Overall, the goal of training a neural network is to learn the relationships between the inputs and the outputs so that the network can accurately predict the output for new, unseen inputs.

Second part:

The perception learning algorithm and the delta learning algorithm are two popular methods used to train feedforward neural networks. Here is a brief discussion of each:

Perceptron Learning Algorithm:

The perceptron learning algorithm is used to train single-layer neural networks that have binary outputs. The goal of this algorithm is to find the optimal weight values that minimize the classification error for a given set of training data. Here are the steps of the perceptron learning algorithm:

1. Initialize the weights to small random values.
2. For each training sample, calculate the output of the network by taking the dot product of the input vector and the weight vector.
3. Apply a threshold function to the ~~input~~ output to produce a binary classification result.

Date
Page

4. If the classification is incorrect, adjust the weights by adding the input vector multiplied by the learning rate and the error (the difference between the target output and the actual output).
5. Repeat steps 2-4 until the error is minimized or a maximum number of iterations is reached.

Delta learning Algorithm:
The delta learning algorithm is used to train multi-layer neural networks that have continous outputs. The goal of this algorithm is to find the optimal weight values that minimize the difference between the network output and the target output for a given set of training data. Here are the steps of the delta learning algorithm:

1. Initialize the weights to small random values.
2. For each training sample, calculate the output of the network by propagating the input forward through the network.
3. Calculate the error between the target output and the actual output.
4. Adjust the weights by adding the input vector multiplied by the learning rate and the error gradient (the derivative of the output with respect to the weights.)
5. Repeat steps 2-4 until the error is minimized or a maximum number of iterations is reached.

Overall, both algorithms are used to adjust the weights of a neural network to minimize the error between the predicted output and the target output. However, the perceptron learning algorithm is used for binary classification problems, while the delta learning algorithm is used for continous output problems.

Date
Page

Q.N.6.

A. Differentiate between NLU and NLG. Why is it difficult to develop a system that could understand natural language? explain.

Ans. NLU (Natural language Understanding) and NLG (Natural Language Generation) are two important aspects of natural language processing (NLP) in AI.

NLU refers to the process of understanding human language by machines. It involves analyzing and interpreting natural language inputs such as text or speech and deriving meaning from them. The main goal of NLU is to enable machines to understand human language and respond appropriately. Examples of NLU include sentiment analysis, text classification, and named entity recognition.

NLG, on the other hand, refers to the process of generating natural language output from structured data or information. It involves converting data or information into human readable text or speech. The main goal of NLG is to enable machines to communicate effectively and naturally with humans. Examples of NLG include chatbots, automated content generation, and summarization systems.

In summary, NLU focuses on understanding human language inputs, while NLG focuses on generating human-like language outputs. Both NLU and NLG are important components of natural language processing and are often used together to create intelligent conversational systems.

Second part:

Developing a system that can understand natural language is a complex and challenging task due to several reasons:

1. Ambiguity: Natural language is often ambiguous, and the

Date
Page

same sentence can have multiple meanings depending on the context. for example, the sentence "I saw her duck" could mean that the speaker saw a bird or the speaker saw a person bending down.

2. Contextual dependencies: The meaning of a sentence can depend on the context in which it is used. for example, the word "bank" can refer to a financial institution or the edge of a river, depending on the context.

3. Variations in Language: Natural language is diverse, and there are many variations in grammer, vocabulary, and syntax across different languages and dialects. This makes it challenging to develop a system that can understand natural language in all its variations.

4. Lack of structure: Natural Language is often unstructured, and there are no clear rules for constructing sentences. This makes it difficult to develop a system that can understand natural language in the same way that humans do.

5. Limited data: To develop a system that can understand natural language, large amounts of training data are required. However obtaining and labeling large amounts of data can be time-consuming and expensive.

6. Common sense reasoning: Understanding natural language often requires common sense reasoning, which is difficult to program into machines. For example, understanding the sentence "John put the cake in the fridge" requires the machine to understand that cakes need to be refrigerated, which is common knowledge for

Date
Page

human but not necessarily for machines.

In summary, developing a system that can understand natural language is challenging due to the ambiguity, contextual dependency, variations in language, lack of structure, limited data, and common sense reasoning required. However, with advancements in machine learning and natural language processing, there has been significant progress in developing systems that can understand and process natural language to some extent.

8. Explain the components and applications of an expert system in AI.

Ans Expert systems are computer-based systems that mimic the problem-solving abilities of a human expert in a particular domain. They are designed to assist human decision-making processes by providing expert-level advice, knowledge, and recommendations. Expert system consists of several components, including:

1. Knowledge base: It is a repository of expert knowledge and domain-specific information. This knowledge base is built by experts in the field and consists of rules, heuristics, and facts.

2. Inference engine: It is the reasoning component of the expert system. It uses the knowledge base to infer new information, make deductions, and provide recommendations.

3. User interface: It is the component that allows the user to interact with the expert system. The user interface can be in the form of a graphical user interface or a natural language interface.

Date
Page

4. Explanation facility : It is the component that provides the user with an explanation of the expert system's reasoning and recommendations.

Some applications of expert systems include:

1. Medical diagnosis : Expert systems can be used in the medical field to assist doctors in diagnosing patients. The system can analyze symptoms, medical history, and other data to provide a diagnosis and treatment recommendations

2. Financial planning : Expert system can assist in financial planning by providing investment recommendations, retirement planning, and tax planning advice

3. Quality control : Expert systems can be used in manufacturing to monitor and control the quality of products. The system can analyze data from sensors and provide real-time feedback to operators.

4. Customer service : Expert system can be used in customer service to provide recommendations and answer questions. The system can analyze customer data and provide personalized recommendations and support.

Overall, expert systems have many applications in various fields and can assist in decision-making processes by providing expert-level advice and knowledge.

Date
Page

Q.N.7.

short notes:

a. State space representation:

State space representation is a modeling technique used in artificial Intelligence (AI) and control systems engineering to represent the possible states of a system or problem, as well as the possible actions that can be taken to move from one state to another. It is essentially a graph or tree structure that represent the different possible states of a system, and the possible transitions between them.

The state space representation consist of a set of nodes representing the different states of the system, and edges or links that connect these nodes and represent the possible transaction or actions that can be taken to move from one state to another. The state space can be used to model a wide variety of systems and problems, ranging from simple puzzles to complex real-world situations.

One of the main advantages of state space representation is that it provides a clear and concise way to model complex systems and problems, and can be used to find optimal solutions to problems through search algorithms such as depth-first search, breadth-first search, and A* search. State space representation is commonly used in fields such as robotics, control systems, and game development among others.

Applications of state space representation include:

- Planning and scheduling systems.
- Robotics and control systems.
- Game development and artificial intelligence.
- Diagonosis and troubleshooting systems.
- Natural language processing and machine translation system.

- Logistics and supply chain management systems.

Overall, state space representation is a powerful modeling technique that is widely used in AI and control systems engineering to represent complex systems and problems, and to find optimal solutions to those problems through search algorithms.

e) Backpropagation:

It is a supervised learning algorithm for training artificial neural networks. It is a widely used algorithm in the field of artifical intelligence and is specifically designed for multi-layer neural networks.

- The backpropagation algorithm uses a supervised learning technique to train the network by propagating errors backwards from the output layer to the input layer. The basic idea is to adjust the weights of the network in order to minimize the difference between the actual output of the network and the desired output.
- The algorithm uses a technique called gradient descent to find the weights that minimize the error between the actual and desired output. During training, the error is propagated backwards through the network and the weights are adjusted based on the error at each layer.
- Backpropagation is known to have some limitations, such as the potential for getting stuck in local minima, slow convergence, and the need for large amounts of data for training. However it remains one of the most popular and effective algorithm for training neural networks, and has been successfully applied to a wide range of applications, including image recognition, speech recognition, and natural language processing.

Class Test (Set A) Saroj Dahal - 191725

Date ______
Page ______

1. Define Artifical intillegence. Discuss the views of AI in computer science. Also mention the application areas of AI.

Ans. Artificial Intillegence (AI) refers to the development of computer systems that can perform tasks that typically require human intelligence, such as visual perception, speech recognition, decision-making and natural language processing. AI is often seen as a subfield of computer science, and involves the study of algorithms, computational models, and cognitive science.

There are different views of AI in computer science. Some researchers see AI as a way to develop intelligent agents that can operate autonomously in complex environments, while other view AI as a means to create cognitive models of human intelligence. Some researchers focus on building systems that can reason and learn, while others focus on developing systems that can interact with humans in natural and intuitive ways.

The application areas of AI are diverse and rapidly expanding. Some of the most popular application areas include:

a. Robotics and automation: AI is used to control and optimize robots in manufacturing and logistics operations.

b. Healthcare: AI is used in medical diagnosis, drug discovery, and personalized treatment planning.

c. Finance: AI is used in fraud detection, risk assessment, and algorithmic trading.

d. Natural language processing: AI is used in chatbots, virtual assistants and speech recognition systems.

e. Image and video recognition: AI is used in facial recognition, object detection, and autonomous vehicles.

f. Gaming: AI is used in game playing, strategy development, and opponent modeling.

9. Education: AI is used in personalized learning, intelligent tutoring systems, and educational data mining.

Overall, AI has the potential to transform many aspect of human life, and its applications are likely to continue expanding in the future.

2. What are the different types of problems? Discuss the necessary steps in problem solving. Explain problem formulation with examples.

ans. In general, there are three main types of problems in AI and computer science:

a. Well-defined problems: These are problems that have a clear and structured set of rules, constraints, and goals. Examples include mathematical problems, logic puzzles, and chess games.

b. Ill-defined problems: These are problems that have vague or incomplete rule, goals, and constraints. Examples include social and political issues, environmental problems, and open-ended creative tasks

c. Transdisciplinary problems: These are problems that require knowledge and expertise from multiple domains, such as engineering, biology, and psychology. Examples include designing a prosthetic limb, developing a self-driving car, and creating an intelligent medical diagnosis system

The necessary steps in problem solving typically include:

a. Problem identification: Clearly define and understand the problem.

b. Problem analysis: Break down the problem into smaller parts and analyze the relationships between them.

c. Solution generation: Develop possible solutions or approaches to the problem.

d. Solution evaluation: Evaluate the possible solutions and select the most appropriate one.

Date
Page

e. Solution Implementation: Implement the solution and monitor its effectiveness.

Problem formulation is the process of transforming an ill-defined problem into a well-defined problem, which can be solved using AI and other computational methods. This involves identifying the goals, constraints, variables, and relevant knowledge and data associated with the problem.

For example, consider the problem of developing a personalized recommendation system for an online shopping platform. The problem formulation process may involve identifying the following:

- Goals: To provide personalized recommendations to users that increase sales and customer satisfaction.
- Constraints: To ensure the recommendations are based on the user's preferences, purchasing history, and demographic information, and comply with privacy and ethical standards.
- Variables: User profiles, product attributes, purchasing patterns, and feedback data.
- Relevant knowledge and data: Machine learning algorithms, data mining techniques, customer behavior models, and market trends analysis.

By formulating the problem in this way, AI and other techniques can be used to develop a solution that addresses the goals and constraints of the problem.

Date ______
Page ______

3. What is the signifnance of search in game playing? Explain Min-Max search strategy with examples. Mention its drawbacks and solution to those drawbacks.

Ans: In game playing, search algorithms are used to find the best possible move in a given game state. The search algorithm starts from the current game state and explores all possible future game states and their corresponding moves. The goal is to find the optimal move that maximizes the player's chances of winning.

Min-max search is a search strategy used in game playing where two players take turns making moves in the game. The goal of the algorithm is to determine the best move for the current player, assuming that the opponent will also play optimally. The algorithm works by exploring the game tree to a certain depth, and then evaluating the utility of the resulting terminal states. The utility is assigned by a heuristic function, which assigns a score to each terminal state based on how good it is for the current player.

For example, consider the game of Tic-Tac-Toe. In the initial ~~tak~~ state, the board is empty. The Min-Max algorithm explores all possible moves for the current player (X) and their corresponding resulting game states. It then evaluates the utility of each resulting state, assuming that the opponent (O) will also play optimally. The algorithm continues to explore the game tree until it reaches a terminal state (win, lose, or draw), and then assigns a utility value to that state.

One drawback of the Min-Max algorithm is that it can be computationally expensive, especially for games with large search spaces. This can be addressed by using techniques such as alpha-beta pruning, which allows the algorithm to eliminate certain branches of the search tree that are guaranteed to be worse than previously explored branches.

In summary, search algorithms such as Min-Max are essential in game playing as they allow for the determination of the best possible move in a given game state. However, the efficiency of these algorithms can be improved through the use of techniques such as alpha-beta pruning.

4. What is the use of logical connectives? List and describe formal logical connectives with truth tables and examples in compound statements.

AN. Logical connectives are symbols used to connect two or more propositions to form a compound statement. They play a crucial role in formal logic and are used in various applications of artifical intelligence such as expert systems, natural language processing and theorem proving.

Here are the formal logical connectives ~~with~~ with their truth tables and examples:

a. Negation (not):

The negation of a proposition p is denoted by ~p, and it is true when p is false and false when p is true.

p	~p.
T	F
F	T

Example: If p represents the proposition, "it is raining" then ~p represents "It is not raining".

Date
Page

b. Conjunction (and):
The conjunction of two propositions p and q is denoted by $p \wedge q$, and it is true only when both p and q are true.

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F.

Example: If p represents the proposition "it is raining" and q represents "I have an umbrella", then $p \wedge q$ represents "It is raining and I have an umbrella".

c. Disjunction (or):
The disjunction of two propositions p and q is denoted by $p \vee q$, and it is true when at least one of the proposition is true.

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F.

Example: If p represents the proposition "it is raining" and q represents "I have an umbrella", then $p \vee q$ represents "it is raining or I have an umbrella".

d. Implication (if-then):

The implication of two proposition p and q is denoted by $p \to q$, and it is false only when p is true and q is false.

p	q	$p \to q$
T	T	T
T	F	F
F	T	T
F	F	T

Example: If p represents the proposition "it is raining". and q represents "I will bring an umbrella", then $p \to q$ represents "if it is raining, then I will bring an umbrella".

e. Bi-implication (if and only if):

The bi-implication of two propositions p and q is denoted by $p \leftrightarrow q$, and it is true only when both p and q have the same truth value.

p	q	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T.

Example: If p represents the proposition "I have an umbrella" and q represents "it is raining", then $p \leftrightarrow q$ represents "I have an umbrella if and only if its is raining".

These logical connectives are used to build more complex propositions and create rules in expert systems. They are also used in

Date
Page

propositional and predicate calculus for theorem proving and in natural language processing for text analysis and classification.

5. "If you send me an e-mail message then I will finish writing the program", "If you do not send me an e-mail message than I will go to sleep early", and "If I go to sleep early then I will wake up feeling refreshed". Lead to the conclusion "If I do not finish writing the program then I will wake up feeling refreshed". Prove by propositional logic.

Ans. Let's assign the following propositions:

p : You send me a email message.

q : I finish writing the program.

r : I go to sleep early.

s : I wake up feeling refreshed

Then we can translate the three given statements into propositional logic: (Hypothesis)

1. $p \rightarrow q$ Conclusion: $\sim q \rightarrow s$

2. $\sim p \rightarrow r$

3. $r \rightarrow s$

S.N.	Steps	Reasons
1.	$p \rightarrow q$	given hypothesis (1)
2.	$\sim q \rightarrow \sim p$	using contrapositive (1)
3.	$\sim p \rightarrow r$	given hypothesis (2)
4.	$\sim q \rightarrow r$	using modus ponens (2) and (3)
5.	$r \rightarrow s$	given hypothesis (3)
6.	$\sim q \rightarrow s$	using modus ponens on (4) and (5).

6. Why do we need predicate logic? Discuss the rule of inference of both propositional logic and predicate logic.

Ans Propositional logic deals with propositions or statements that can be either true or false, and uses logical operator like AND, OR, NOT, and IF-THEN to form compound propositions. However, propositional logic cannot handle more complex statement that involve quantification and relations between objects, which are common in natural language and many areas of knowledge representation. This is where predicate logic comes in.

Predicate logic, also known as first-order logic, extends propositional logic by introducing quantifiers like FOR ALL ($\forall$) and THERE EXISTS ($\exists$) to express statements about objects and their properties, and relations between them. It allows for more complex statements to be represented such as natural language processing, expert system, and automated reasoning.

In propositional logic, some common inference rules include Modus ponens, Modus Tollens, Conjunction, Disjunction, and Contradiction. These rules operate on propositions as a whole, without taking into account the internal structure of propositions.

Name	Inference rule.
P $P \rightarrow Q$ $\therefore P$	Modus ponens (mode that affirms)
$\sim Q$ $P \rightarrow Q$ $\therefore \sim P$	Modus tollens (mode that denies)
$P \rightarrow Q$ $Q \rightarrow R$ $\therefore P \rightarrow R$	hypothetical syllogism.

Date
Page

$p \vee q$ disjunctive syllogism.
$\sim p$
$\therefore q$

p addition.
$\therefore p \vee q$

$p \wedge q$ simplification.
$\therefore p$

p conjunction.
q
$\therefore p \wedge q$

$p \vee q$ resolution.
$\sim p \vee r$
$\therefore q \vee r$

Whereas, In predicate logic, there are additional inference rules that take into account the internal structure of statements, such as Universal Instatiation, Existential Generalization, and Universal Generalization. These rules allow for the manipulation of quantifiers and predicates, and the derivation of more complex statements.

Rule of Inference	Name
$\forall x\, P(x)$ $\therefore P(c)$	Universal instantiation.
$P(c)$ for an arbitrary c $\therefore \forall x\, P(x)$	Universal generalization.

Date
Page

$\exists x\, P(x)$ — Existential instantiation.

$\therefore P(c)$ for some element c

$P(c)$ for some element c — Existencial generalization.

$\therefore \exists x\, P(x)$

7. Represent the following statement into predicate logic:
   a. Subit and shrabin are players.
   b. All gardeners love the sun.
   c. kiran has at least one sister.
   d. All purple mushrooms are poisonous.

Ans. a. Let $P(x)$ be "x is player", then the statement can be represented as $P(subit) \wedge P(shrabin)$.

b. Let $G(x)$ be "x is gardener" and $S(x)$ be "x loves the sun", then the statement can be represented as $\forall x\,(G(x) \rightarrow S(x))$.

c. Let $S(x)$ be "x is a sister of kiran", then the statement can be represented as $\exists x\,(Sx)$.

d. Let $M(x)$ be "x is a mushroom" and $P(x)$ be "x is purple and poisonous", then the statement can be represented as $\forall x\,((M(x) \wedge P(x)) \rightarrow P(x))$.

8. Given expression:

All men are mortal. Einstein is a man. Prove that "Einstein is mortal" using FOPL.

Ans. To prove "Einstein is mortal" using FOPL, we can use the following steps:

Step 1 : Define the domain of discourse.
Let the domain of discourse be all individuals.

Step 2 : Define the predicates and constants.
Let the predicate $M(x)$ be "x is a man".
Let the predicate E be "Einstein".
Let the predicate $T(x)$ be "x is mortal".
Let the predicate

Step 3 : Write the premises in predicate logic.
"All men are mortal" can be written as:

$$\forall x\,(M(x) \rightarrow T(x))$$

"Einstein is a man" can be written as:

$$M(E)$$

Step 4 : Apply Universal Instantiation.
From the premise $\forall x(M(x) \rightarrow T(x))$, we can instantiate x to Einstein to get:

$$M(E) \rightarrow T(E).$$

Step 5 : Apply Modus Ponens.
Using the premise $M(E)$ and the result of step 4, we can apply Modus Ponens to derive: $T(E)$.

Therefore, we have proven "Einstein is mortal" using FOPL.

9. Define a conjunctive normal form. Converting to conjunctive normal form:

$$(P \wedge \sim q) \vee (\sim r \wedge q) \text{ and } \sim((\sim p \rightarrow \sim q) \wedge \sim r).$$

Ans A conjunctive normal form (CNF) is a specific form of a propositional logic formula that is expressed as a conjunction of clauses, where each clause is a disjunction of literals i.e.,

Date ______
Page ______

a variable or it's negation). In other words, a CNF formula is a logical expression that is constructed by joining multiple clauses using "AND" operators, where each clause contains one or more literals joined together using "OR" operators. A formula is said to be in CNF if it satisfies these conditions:

1. The formula is a conjunction of one or more clauses
2. Each clause is a disjunction of one or more literals.
3. No literal appears more than once in any clause

CNF is a useful form for logic formulas because it can be easily implemented in hardware and software, and many efficient algorithms exists for testing ~~stuck~~ satisfiability and performing logical operations on CNF formulas.

→ To convert $(p \wedge \sim q) \vee (\sim r \wedge q)$ to conjunctive normal form, we can follow these steps:

1. Distribute the $\vee$ over the $\wedge$ using the distributive law:
   $(p \vee \sim r) \wedge (p \vee q) \wedge (\sim q \vee \sim r) \wedge (\sim q \vee q)$
2. Simplify the expressions $(\sim q \vee q)$ to true and $(p \vee \sim r) \wedge (p \vee q)$ to $p \vee (\sim r \wedge q)$. So the conjuctive normal form of $(p \wedge \sim q) \vee (\sim r \wedge q)$ is:
   $(p \vee \sim r) \wedge (p \vee q) \wedge (\sim q \vee \sim r)$.

→ To convert $\sim((\sim p \to \sim q) \wedge \sim r)$ to conjunctive normal form, we can follow these steps:

1. Use the equivalence $\sim(p \to q) \equiv p \wedge \sim q$ to get:
   $\sim((\sim p \wedge q) \wedge \sim r)$.
2. De Morgan's Law $\sim(A \wedge B) \equiv \sim A \vee \sim B$:
   $((\sim\sim p \vee \sim q) \vee r$.
3. Double negation $\sim\sim p \equiv p$
   $(p \vee \sim q) \vee r$.

So the conjunctive normal form of $\sim((\sim p \to \sim q) \wedge \sim r)$ is:
$(p \vee \sim q \vee r)$

Date
Page

10. How is the resolution algorithm used as a rule of inference in predicate logic? "All over smart people are stupid. Children's of all stupid persons are naughty. Roney is a children of Harry. Harry is over smart". Prove that "Roney is naughty" using a resolution algorithm.

Ans. The resolution algorithm is a rule of inference in predicate logic that is used to determine the validity of a statement by negating it and attempting to derive a contradiction from the resulting clauses. Here's how we can use the resolution algorithm to prove that "Roney is naughty" based on the given statements:

1. Translate the given statements into predicate logic using variables and predicates:
   - $\forall x$ (oversmart(x) $\rightarrow$ stupid(x)).
   - $\forall x \forall y$ (stupid(x) $\wedge$ parent(y, x) $\rightarrow$ naughty(y)).
   - parent(Harry, Roney).
   - Oversmart(Harry).

2. Negate the statement we want to prove (i.e. $\sim$naughty(Roney)) and convert it into clause form:
   - naughty(x)
   - $\sim$naughty(Roney).

3. Apply resolution to the clauses by resolving on the negated predicate and the second statement, which gives us the new clause $\sim$naughty(Harry):
   - $\sim$naughty(Roney), $\forall x \forall y$ (stupid(x) $\wedge$ parent(y,x) $\rightarrow$ naughty(y)).
   - $\sim$naughty(Harry), $\forall x$ (oversmart(x) $\rightarrow$ stupid(x)).

Date
Page

4. Apply resolution again to the new clause and the first statement, which gives us the empty clause (i.e. a contradiction):
- ~ naughty (Harry), ∀x (oversmart (x) → stupid (x)).
- stupid (Harry) (resolved from 2 and 3).
- oversmart (Harry) → stupid (Harry) (resolved from 1 and 3).
- ~ oversmart (Harry) (derived by modus tollens from 4).
- naughty (Roney) (resolved from 2 and 4).
- ~ naughty (Roney), naughty (Roney) (resolved from 2 and 5).

Since we have derived a contradiction, we can conclude that our initial assumption (~ naughty (Roney)) must be false, and therefore "Roney is naughty" is true.

11. What is the Bayesian Network? Explain how Bayesian Network represents and inference the uncertain knowledge with an appropriate example.

Ans. Bayesian Network is a probabilistic graphical model used for representing and reasoning about uncertain knowledge. It consists of a set of variables and their conditional dependencies represented as a directed acyclic graph (DAG). Each node in the DAG represents a random variable and the directed edges between them represent the probabilistic dependencies between the variables.

Bayesian Networks are useful in a wide range of applications, such as medical diagnosis, credit risk analysis, speech recognition, natural language processing, and many more.

An example of a Bayesian Network is a medical diagnosis system. Suppose a patient comes to a doctor with some symptoms, and the doctor needs to diagnose the underlying disease. The symptoms can be represented as variables, and the disease can be represented as a hidden variable that causes the observed

Date
Page

symptoms. The Bayesian Network can then be used to represent the conditional probabilities of the variables given their parent variables.
For instance, let's say that the patient is experencing fever, cough, and fatigue. These symptoms can be represented as variables F, C, and T respectively. So, the network has three nodes:

1. Fever (F): Indicates whether the patient is experiencing fever or not.
2. Cough (C): Indicates whether the patient is experiencing cough or not.
3. Fatigue (F): Indicates whether the patient is experiencing fatigue or not.
4. Flu.
5. Cold
6. Pneumonia.

The dependencies among these variables can be represented by a Bayesian network, which is a directed acyclic graph (DAG) where each node represents a variable and each edge represents a probabilistic dependency between the variables.

For example, the network might have the following structure:

Fever
↓
Flu, Cold
↓
Cough, Fatigue
↓
Pneumonia.

This structure represents the fact that if the patient has the flu or a cold, they are more likely to experience a cough and fatigue. If the patient has pneumonia, they are more likely to experience a fever, cough, and fatigue.

Using the Bayesian network, we can perform probabilistic inference to determine the probability that the patient has each of the possible diagnoses given their symptoms. For example, if the patient has a fever, cough, and fatigue, we can use the network to calculate the probability that they have flu, a cold, or pneumonia.

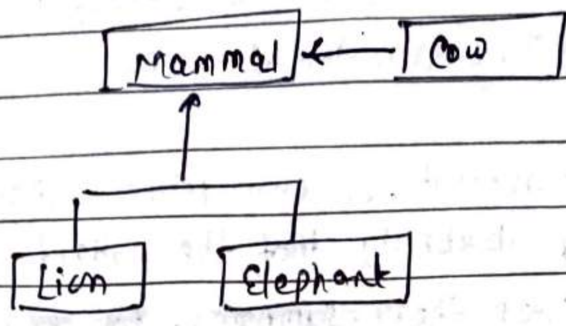
The Bayesian network allows us to represent and reason about uncertain knowledge by explicitly modeling the probabilistic dependencies between variables. This makes it a useful tool for medical diagnosis system, where the diagnosis of a patient is often uncertain and depends on a variety of factors.

12. What are the different knowledge representations? Explain the semantic network with examples.

Ans. There are several different knowledge ~~requires~~ representations in artificial intelligence, including semantic networks, frames, rules and ontologies. Each representation has its own strengths and weakness and is suited for different types of problems.

A semantic network is a graphical representation of knowledge that shows how different concepts are related to each other. It consists of nodes (or vertices) that represent concept and edges (or arcs) that represent relationships between the concepts. The nodes can also have attributes that describe their properties.

For example, consider the following semantic network for the domain of animals:



In this network, the nodes represent concept such as Mammal, Cow, Lion, and Elephant. The edges represent relationships between the concepts, such as "is-a" or "part-of". For example, the edge from Cow to Mammal indicates that a Cow is a type of Mammal. The network also shows that Lion and Elephant are also types of Mammal.

Semantic networks can be used for various tasks such as knowledge representation, reasoning, and natural language understanding. They can also be used to build expert systems or to model human decision-making processes.

13. Write short note on : (any one).

- Production Rule system:

A production rule system (PRS), also known as a production system or a rule-based system, is a type of knowledge representation and reasoning system in artificial intilligence. It consist of a set of production rules, which are conditional statements that link given conditions to actions or conclusions.

The rules in a PRS are typically in the form of "if-then" statements, where the "if" part specificies the conditions that need to be met for the rule to be applied, and the "than" part specifies the action or conclusion that should be taken. A PRS uses a set of working memory elements that represent the current state of the system, and a control strategy that selects which rule to apply based on the current state and the rule conditions.

Production rule systems have been used in various applications, such as expert systems, diagnosis and planning systems, natural language processing, and machine learning. They provide a flexible and modular way of representing and reasoning about knowledge, and can be easily modified or extended by adding new rules or modifying existing ones.

Example of a production rule in a diagnostic system for a car:
IF there is smoke coming from the engine AND the engine is running rough THEN there may be problem with the engine's fuel system.

In this example, the rule specifies the conditions that indicate a problem with the car's fuel system based on the symptom observed (smoke and rough engine) and triggers an action or conclusion (investigate the fuel system) if those conditions are met.

Saroj Dahal - 191725

Class Test (Set 2).

Date
Page

1. Explain the Turing test with its basic requirements. Machines can be made intelligent artifically but ultimately humans make the machines. So who is more intelligent - the artificial machine or the person? Discuss.

Ans. The Turing Test is a test proposed by Alan Turing in 1950 to determine a machines ability to exhibit human like intilligence. The basic requirement for the test are:

a. A human judge communicates with two entities, one human and one machine, through a text-based interface.

b. The judge does not know which entity is the machine and which is the human.

c. The judge engages in a natural language conversation with both entities.

d. If the judge cannot determine which entity is the machine and which is the human, then the machine is said to have passed the Turing test.

The Turing test has been a subject of debate among AI researchers, philosophers and psychologists. Some argue that it is an effective way to measure a machine's intilligence, while others argue that it is not a sufficient measure of true intelligence.

As for the question of whether a machine or a human is more intilligent, it depends on how one defines intilligence. Machines can perform certain tasks with greater speed and accuracy than humans, but they lack the creativity, emotional intilligence, and adaptability that humans posses. Humans also have a deeper understanding of the world around them and the ability to reason abstractly, whereas machines can only operate within the limitations of their programming.

Date
Page

In conclusion, while machines can be intilligent in certain areas, they cannot be considered more intelligent than humans as they lack the breadth and depth of human intelligence.

Q.N.2.

- Give examples of deterministic and non deterministic problem space. You are given two jugs ; a 4 liter gallon one and a 3 liter gallon one. Neither have any measuring markers on it. There is a pump that can be used to fill the jugs with water. How can you get exactly 2 liters of water into the 4 liters galloon jug? Solve this problem with state space representation.

Ans. Deterministic problems have a predictable outcome based on a given set of rules or actions. Examples of deterministic problem spaces include:

1. Chess : Each move made by a player has predetermined outcome.
2. Rubik's Cube : The movement of each cube is based on a set of rules that have a predetermined outcome.
3. Sudoku : The placement of each number in a Sudoku puzzle is based on a set of rules and has a pre-determined outcome.

Non-deterministic problems do not have a predictable outcome based on a given set of rules or actions. Examples of non-deterministic problem space include:

1. Traveling Salesman problem : Finding the shortest route between a set of cities is a non-deterministic problem as there are multiple possible routes and the outcome is unpredictable.
2. Chess Endgame : In some chess endgame scenarios, the outcome is unpredictable due to the number of possible moves available.

Date
Page

3. Maze solving : In a complex maze, finding the correct path to the destination is non-deterministic as there are multiple possible paths and the outcome is unpredictable.

Solution :
we firstly define the problem as state space search and set the ordered pair of integers (x, y). Where x is the quantity of water in a 4L jug and y is the quantity of water in a 3L jug.
The start state is (0,0).
The final state is (2,n).
We also define some rules from state space search to solve this problem. Here is the solution.

Water quantity in 4L jug.	Water quantity in 3L jug.	Rule Applied
0	0	2
0	3	10
3	0	2
3	3	7
4	2	4
0	2	10
2	0.	

Q.N.3.
What is the concept of MIN-MAX searching technique? Explain alpha-beta pruning by taking an appropriate example.

Ans. MIN-max algorithm:
It is a recursive algorithm for choosing the next move in an player game, usually a two player game.
• The value is computed by means of a position evaluation

Date
Page

function and it indicates how good it would be before a player reaches the position.

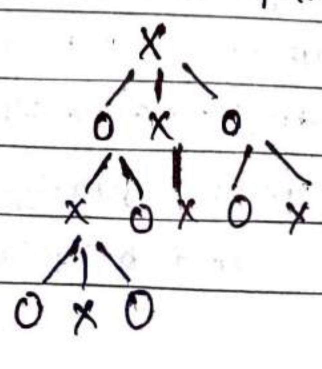
- The player then makes the move that maximizes the minimum value of the position from the opponent's possible moves called maximizing player and other player minimize the maximum value of the position called minimizing player.

We first consider games with two players; MAX and MIN. MAX moves first, and then they take turns moving until the game is over. Each level of the tree alternates. MAX is trying to maximize score, and MIN is trying to minimize MAX score in order to undermine success. We first consider games with two players; MAX and MIN. MAX move first, and then they take turns moving until the game is over. Each level of the tree alternates, MAX is trying to maximize score, and MIN is trying to minimize MAX score in order to undermine success.

Second part:

Alpha-beta pruning is a search algorithm used for minimizing the number of nodes that are evaluated by the minmax algorithm in a game tree. It works by removing parts of the game tree that are guaranteed not to influence the final result.

Lets take an example of tic-tac-toe game to explain alpha-beta pruning. Consider the following game tree for tic-tac-toe.

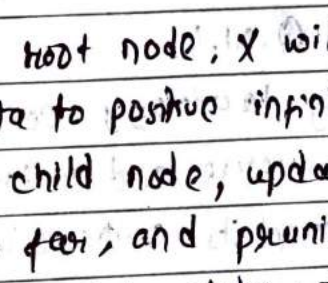


Date
Page

In this game tree, X is the maximizer and O is the minimizer. X wants to maximize the outcome of the game, while O wants to minimize it.
With alpha-beta pruning, we can eliminate some of the branches of the game tree without having to evaluate every node. The algorithm maintains two values: alpha, which represents the maximum value that X can guarantee, and beta, which represents the minimum value that O can guarantee.

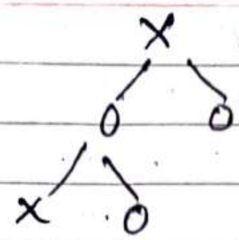
Starting at the root node, X will initially set alpha to negative infinity and beta to positive infinity. Then, X will recursively evaluate each child node, updating alpha with the maximum value found so far, and pruning any nodes with a value less than or equal to alpha. If a node has a value greater than or equal to beta, then O will never choose this node, so we can prune this node and all of its descendants.

Let's see an example. Consider the following subtree of the game tree:



Starting at the root node, X sets alpha to negative infinity and beta to positive infinity. X evaluates the left child node and updates alpha to 0, since X can guarantee at least a draw from this node. Then, X evaluates the right child node, which has a value of -1. Since this value is less than or equal to alpha, X prunes this node and all of its descendants. The resulting pruned subtree is:

Date ______
Page ______



Using alpha-beta pruning, we have eliminated a significant portion of the game tree without affecting the final result of the game.

Q.N.4.

What do you mean by tautology, contradiction and contingency? Mention the inverse, converse and contrapositive of an implication. Prove that implication and its contrapositive are logically equivalent.

AN. In propositional logic, a tautology is a statement that is true in all possible interpretations, a contradiction is a statement that is false in all possible interpretations, and a contingency is a statement that is neither a tautology nor a contradiction.

The inverse of an implication "$p \to q$" is "$\sim p \to \sim q$"
The converse is "$q \to p$" and contrapositive is "$\sim q \to \sim p$".

To prove that an implication and its contrapositive are logically equivalent, we can use a truth table. Let "p" and "q" be propositions then;

p	q	$p \to q$	$\sim q$	$\sim p$	$\sim q \to \sim p$	$p \to q \equiv \sim q \to \sim p$
T	T	T	F	F	T	T
T	F	F	T	F	F	T
F	T	T	F	T	T	T
F	F	T	T	T	T	T

Date
Page

As we can see from truth table, the columns for "$p \to q$" and "$\sim q \to \sim p$" have the same truth values for all possible combinations of truth values for "p" and "q". Therefore, we can conclude that an implication and its contrapositive are logically equivalent.

Q.N.S.

"It is not sunny this afternoon and it is colder than yesterday", "We will go swimming only if it is sunny", "If we do not go swimming then we take a canoe trip", and "If we take a canoe trip then we will be home by sunset" leads to the conclusion "We will be home by sunset". Prove by using propositional Logic.

Ans. let the given hypothesis be represented by proposition variables such that:

p = It is sunny this afternoon.

q = It is colder than yesterday.

r = We will go swimming.

s = We will take a canoe trip.

t = We will be home by sunset.

Given: $\sim p \wedge q$, $r \to p$, $\sim r \to s$ and $s \to t$; Conclusion : t

Steps	Reasons:
a. $\sim p \wedge q$	Hypothesis.
b. $\sim p$	Simplification using step (a).
c. $r \to p$	Hypothesis.
d. $\sim r$	Modus Tollens using steps (b) and (c).
e. $\sim r \to s$	Hypothesis.
f. s	Modus ~~Tolle~~ Ponens using step (d) and (e)
g. $s \to t$	Hypothesis
h. t	Modus ponens using steps (e) and (f)

Date
Page

Q.N.6.

What is the use of quantifiers? Discuss and differentiate between universal and existencial quantification.

→ In predicate logic, quantifiers are used to specify the scope of a variable in a statement. Instead of just specifying quantification, from quantifiers we can wholly represent the state in a particular domain in just once mentionings. There are two types of quantifiers and they are: Universal quantifiers and Existencial quantifier.

Universal quantification, denoted by the symbol $\forall$ (pronounced "for all"), asserts that a certain predicate is true for every possible value of the variable. For example, the statement $\forall x (x > 0)$ means "For all x, x is greater than zero". This statement is true because every possible value of x that can be plugged into the predicate results in a true statement.

Existential quantification, denoted by the symbol $\exists$ (pronounced "there exists"), asserts that there is at least one individual that satisfies a certain predicate or condition. For example, the statement $\exists x (x > 0)$ means "There exists an x such that x is greater than zero". This statement is true because at least one positive number exists.

The difference between universal and existential quantification is in the scope of the statement. Universal quantification is a statement that applies to all possible values of a variable, whereas existencial quantification asserts the existence of at least one value that satisfies the predicate.

In logical terms, a statement with universal quantification is equivalent to the negation of a statement with existential quantification, and vice-versa. For example, the statement $\forall x P(x)$ is equivalent to $\sim \exists x \sim P(x)$, and the statement $\exists x P(x)$ is equivalent to $\sim \forall x \sim P(x)$.

Q.N.7

Represent the following statement into predicate logic.

a. All students are smart :

$\forall x\,(Student(x) \rightarrow Smart(x))$

b. Every person loves themselves :

$\forall x\,(Person(x) \rightarrow Loves(x,x))$

c. No student likes every lecture.

$\forall x \forall y\,(Student(x) \wedge Lecture(y) \rightarrow \sim Likes(x,y))$

d. Everyone is a friend of someone.

$\forall x \exists y\,(Friend(x,y))$.

Q.N.8.

Given expression: "Lions are dangerous animals" and "There are lions". Prove that "There are dangerous animals" using FOPL.

→ The given expressions can be represented in first-order predicate logic (FOPL) as follows:

$D(x)$: x is a dangerous animal.

$L(x)$: x is a lion.

T : there exists something.

Date ______
Page ______

The given expressions can be translated as:

$\forall x (L(x) \rightarrow D(x))$: Lions are dangerous animals

$\exists x\, L(x)$: There are lions.

To prove that "There are dangerous animals" we need to show that there exists an animal which is both a lion and dangerous. This can be done by using existential instantiation and modus ponens.

Let's assume that there exists an animal y which is a lion. So we can write:

$L(y)$ (Existential instantiation).

Using modus ponens, we can then conclude that y is a dangerous animal:

Date
Page

Q.N.9.

Define a disjunctive normal form. Convert to disjunctive normal form for $(p \to q) \to (\sim r \wedge q)$.

Ans A disjunctive normal form (DNF) is a way of expressing a logical formula as a disjunction of conjunctive clauses. It is also known as a sum of products form. In DNF, the formula is written as a logical OR (disjunction) of logical AND (conjunction) of literals (propositional variables of their negations. DNF can be used to simplify logical formulas and make them easier to evaluate.

Given: $(p \to q) \to (\sim r \wedge q)$.

To convert given expression into disjunctive normal form, we can follow these steps:

1. Rewrite the implication as an OR of the negation of the antecedant and the consequent:

$$\sim(p \to q) \vee (\sim r \wedge q)$$

2. Use De Morgan's law to move the negation inside and distribute the conjunction:

$$(\sim\sim p \wedge \sim q) \vee (\sim r \wedge q)$$

$$(p \wedge \sim q) \vee (\sim r \wedge q).$$

3. Apply the distributive law to get the disjunctive normal form:

$$(p \vee \sim r) \wedge (p \vee q) \wedge (\sim q \vee \sim r).$$

Date
Page

1. Q.N.10.

Explain the principle of resolution in predicate logic with an appropriate example. Also describe the steps involved in the process.

Ans. The principle of resolution is a powerful inference rule used in predicate logic to derive new logical statements or to prove the validity of an argument. It is based on the idea of reducing a complex statement to a simpler one by finding a common term that can be eliminated using the process of unification. The resulting statement is known as the resolvent.
Here is an example that illustrates the principle of resolution:
Consider the following two statements:
1. All dogs bark.
2. Fido is a dog.

We can use the principle of resolution to derive the conclusion that "Fido barks". The steps involved in this process are as follows:

Step 1: Convert the statements into logical form using predicates and variables:

1. $\forall x\,(Dog(x)) \rightarrow Bark(x))$
2. $Dog(Fido)$.

Step 2: Negate the conclusion and convert it into logical form:

3. $\sim Bark(Fido)$.

Step 3: Apply the resolution rule by finding a common term between the negation of the conclusion and one of the premises. In this case, the common term is "Dog (Fido)".

4. Premise 2: Dog (Fido)
5. From premise 1 and 4, we can infer Bark (Fido).

Date
Page

6. From 3 and 5, we can infer a contradiction ($\sim$Bark(Fido) $\wedge$ Bark(Fido))

Since we have derived a contradiction, we can conclude that the original premises entail the conclusion that "Fido barks". The principle of resolution is a fundamental technique used in automated theorem proving and is the basis of many logic programming languages such as prolog. It is also a key component in natural language processing and knowledge representation.

Q.N.11

What is the probability that a patient has dengue with a fever?
Given data: A doctor is aware that disease dengue causes a patient to have a fever, and it occurs 60% of the time. He is also aware of some more facts, which are given as follows:
The known probability that a patient has dengue disease is 1/50,000. The known probability that a patient has a fever is 5%.

Ans. To find the probability that a patient has dengue with a fever, we can use Baye's Theorem.
Let D be the event that a patient has dengue and F be the event that a patient has a fever.

We are given:
$P(D) = 1/50,000$ (Probability that a patient has dengue).
$P(F) = 0.05$ (probability that a patient has a fever).
$P(F|D) = 0.6$ (Probability that a patient with dengue has a fever)

We want to find $P(D/F)$ (Probability that a patient has dengue given that they have a fever).

$$P(D/F) = \frac{P(F/D) \cdot P(D)}{P(F)}$$

$$= \frac{0.6 \times 1/50,000}{0.05}$$

$$= 0.00024$$

Q.N.12.

Differentiate between semantic networks and frames. Mention examples of each.

Ans. Semantic networks and frames are two different knowledge representation techniques used in artificial intelligence. Semantic networks are a graphical representation of knowledge that consists of nodes or vertices connected by labeled arcs or edges to show relationships between the nodes. In other words, it represents concepts or entities and their relationships with other concepts or entities. For example, a semantic network can represent the concept of "animals" and their relationship with other concepts such as "mammals", "birds", "reptiles", "insects" etc.

On the other hand, frames are a more structured knowledge representation technique that represents objects or concepts as frames or structures consisting of slots or attributes and their values. Frames are used to represent knowledge in terms of classes and instances, where a class represents a set of objects or concepts that share common attributes or characteristic and an instance is an individual member of that class. For example, a frame for a "car" might have slots such as "make", "model", "year", "color", "engine type", "fuel type" etc.

Date ____
Page ____

Examples of semantic networks include WordNet, which represents the meanings of words and their relationships with other words, and ConceptNet, which represents general knowledge about the world. Examples of frames include the semantic Web, which represents web resources and their relationships, and the Cyc knowledge base, which represents a large amount of general knowledge.

Q.N.13.

Write short note on (any one):

- Means-End Analysis:

Means-End Analysis is a problem-solving strategy used in artifical intillegence and cognitive psychology. It involves breaking down a complex problem into smaller sub-problems and finding the means to reach the desired end goal.

The basic principle behind Means-End Analysis is to identify the difference between the current state of the problem and the desired goal state, and then working towards reducing that difference by taking appropriate actions.

The steps involved in Means-End Analysis are as follows:

1. Define the problem and identify the end goal.
2. Analyze the current state of the problem and identify the differences between the current state and the goal state.
3. Generate a set of sub-goals that can be achieved to reduce the differences.
4. Develop a plan of actions to achieve the sub-goals.
5. Execute the plan and monitor progress toward the goal.
6. Repeat the process until the goal is achieved.

MEA is often used in expert systems, robotics, and planning and scheduling applications. It is also used in natural language processing and decision making system.