



SAMSKRUTI COLLEGE OF ENGINEERING & TECHNOLOGY

(Approved by AICTE, New Delhi & Affiliated to JNTUH.)

Kondapur(V), Ghatkesar(M), Medchal(Dist)



Course Hand Out

Subject Name: Artificial Intelligence

Prepared by: V.Chandrasekhar, Associate Professor, CSE

Year, Semester, Branch, Regulation: III Year B.Tech., II-Sem, CSE, R-16

UNIT -I

KEY POINTS:

Artificial Intelligence (AI) is a branch of *Science* which deals with helping machines find solutions to complex problems in a more human-like fashion.

Importance of AI :

Game Playing

Speech Recognition

Understanding Natural Language

.

Computer Vision

Expert Systems

Heuristic Classification

The applications of AI :

- a) Consumer Marketing
- b) Identification Technologies
- c) ID cards e.g., ATM cards
- d) Intrusion Detection
- e) Machine Translation

Heuristics

A heuristic is a way of trying to discover something or an idea imbedded in a Program . The term is used variously in AI. *Heuristic functions* are used in some Approaches to search to measure how far a node in a search tree seems to be from a Goal. *Heuristic predicates* that compare two nodes in a search tree to see if one is better than the other, i.e. constitutes an advance toward the goal, may be more useful.

Search and Control Strategies :

Searching is the universal technique of problem solving in AI. There are some single-player games such as tile games, Sudoku, crossword, etc.,. The search algorithms help you to search for a particular position in such games.

Search Terminology :

- **Problem Space** – It is the environment in which the search takes place. (A set of states and set of operators to change those states)
- **Problem Instance** – It is Initial state + Goal state.
- **Problem Space Graph** – It represents problem state. States are shown by nodes and operators are shown by edges.
- **Depth of a problem** – Length of a shortest path or shortest sequence of operators from Initial State to goal state.
- **Space Complexity** – The maximum number of nodes that are stored in memory.
- **Time Complexity** – The maximum number of nodes that are created.
- **Admissibility** – A property of an algorithm to always find an optimal solution.
- **Branching Factor** – The average number of child nodes in the problem space graph.
- **Depth** – Length of the shortest path from initial state to goal state.

Problem Solving :

Problem solving is an important aspect of Artificial Intelligence. A problem can be considered to consist of a goal and a set of actions that can be taken to lead to the goal. At any given time, we consider the state of the search space to represent where we have reached as a result of the actions we have applied so far.

Uniformed or Blind search :

Breadth First Search (BFS) :

BFS expands the leaf node with the lowest path cost so far, and keeps going until a goal node is generated. If the path cost simply equals the number of links, we can implement this as a simple queue (“first in, first out”).

Depth First Search (DFS):

DFS expands the leaf node with the highest path cost so far, and keeps going until a goal node is generated. If the path cost simply equals the number of links, we can implement this as a simple stack (“last in, first out”).

Greedy Hill Climbing :

This is a form of hill climbing which chooses the node with the least estimated cost at each state. It does not guarantee an optimal solution path. Greedy hill climbing is susceptible to the problem of getting stuck at a local optimum. The Greedy Hill Climbing Algorithm

- 1) Evaluate the initial state.
 - 2) Select a new operator.
 - 3) Evaluate the new state
 - 4) If it is closer to the goal state than the current state make it the current state.
 - 5) If it is no better ignore
 - 6) If the current state is the goal state or no new operators are available, quit.
- Otherwise repeat steps 2 to 4.

What are Agent and Environment?

An **agent** is anything that can perceive its environment through **sensors** and acts upon that environment through **effectors**.

- A **human agent** has sensory organs such as eyes, ears, nose, tongue and skin parallel to the sensors, and other organs such as hands, legs, mouth, for effectors.
- A **robotic agent** replaces cameras and infrared range finders for the sensors, and various motors and actuators for effectors.
- A **software agent** has encoded bit strings as its programs and actions.

Properties of Environment :

The environment has multifold properties –

- **Discrete / Continuous** – If there are a limited number of distinct, clearly defined, states of the environment, the environment is discrete (For example, chess); otherwise it is continuous (For example, driving).
- **Observable / Partially Observable** – If it is possible to determine the complete state of the environment at each time point from the percepts it is observable; otherwise it is only partially observable.
- **Static / Dynamic** – If the environment does not change while an agent is acting, then it is static; otherwise it is dynamic.
- **Single agent / Multiple agents** – The environment may contain other agents which may be of the same or different kind as that of the agent.
- **Accessible / Inaccessible** – If the agent's sensory apparatus can have access to the complete state of the environment, then the environment is accessible to that agent.
- **Deterministic / Non-deterministic** – If the next state of the environment is completely determined by the current state and the actions of the agent, then the environment is deterministic; otherwise it is non-deterministic.
- **Episodic / Non-episodic** – In an episodic environment, each episode consists of the agent perceiving and then acting. The quality of its action depends just on the episode itself. Subsequent episodes do not depend on the actions in the previous episodes. Episodic environments are much simpler because the agent does not need to think ahead.

Important Questions from JNTUH Previous Question Papers :

- | | |
|--|---|
| 1. Define Heuristic Function? | 2 |
| 2. What is space space search? | 2 |
| 3. What are the problems underlying with AI? | 2 |
| 4. Give brief note in Intelligent systems? | 3 |
| 5. Differentiate BFS and DFS | 3 |

6. What is AI? What are its elements? 5
7. Explain the Heuristic search techniques? 5
8. List and explain the applications of AI 5
9. Discuss the characteristics of AI problem? 5
10. Give a brief note on Alpha-Beta Pruning? 5

11. What are constraint satisfaction problem? How can you formulate them as search problem? 10
12. What is Heuristic search? Explain in detail about iterative deepening A* algorithm? 10
13. Define Heuristic search? What are the advantages of Heuristic search? 8
14. Describe the mini max algorithm with example? 8
15. Explain the state space representation of Water-Jug Problem 8

1. A graph with no edges connecting the adjacent vertices is _____
2. A graph with edges connecting the adjacent vertices is _____
3. A graph with out any directions connecting the adjacent vertices is _____
4. A graph with directions to all of it's adjacent vertices is _____
5. A graph with weights associated on edges is _____
6. A graph with out any weights associated on edges is _____
7. The initial vertex of a graph or tree is called _____

8. The Final vertex of a graph or tree is called_____
9. Heuristic is a technique that _____of a search process.
10. The representation of knowledges has multiple_____ associated with it.

UNITS –II & III

Constraint Satisfaction Problem (CSP) :

Search can be used to solve problems that are limited by constraints, such as the eight queen's problem. Such problems are often known as Constraint Satisfaction Problems or CSPs.

In this problem, eight queens must be placed on a chess board in such a way that no two queens are on the same diagonal, row, or column.

This kind of problem is known as a constraint satisfaction problem (CSP) because a solution must be found that satisfies the constraints.

A goal node in this tree is one that satisfies the constraints that no two queens can be on the same diagonal, row, or column.

An extremely simplistic approach to solving this problem would be to analyze every possible configuration until one was found that matched the constraints.

Rules to solve Cryptarithmic Questions

1. Each letter should have a unique and distinct value.
2. Each letter represents only one digit throughout the problem.
3. Numbers must not begin with *zero*. i.e. 0937 (wrong), 937 (correct).
4. You have to find the value of each letter in the Cryptarithmic.
5. There must be only one solution to the problem.
6. After replacing letters by their digits, the resulting arithmetic operations must be correct.

So, the problem is to find the unique digit corresponding to a unique letter.

Explanation:

```
  SEND
+ MORE
-----
 MONEY
-----
```

From the given data, the value of M will be 1 because it is the only carry-over possible from the sum of 2 single digit numbers in column 4. $M = 1$, $S + 1 = a$ (two digit number). So $S = 1$ and O cannot be 1 but 0. Also E and N are consecutive. With trial and error possibilities, we get $SEND = 9567$, $MORE = 1085$ and $MONEY = 10652$.

This cryptographic problem can be solved by using a Generate and Test method,

applying the following constraints:

- a) Each letter represents exactly one number.
- b) No two letters represent the same number.

Generate and Test is a brute-force method, which in this case involves cycling through all possible assignments of numbers to letters until a set is found that meets the constraints and solves the problem.

Minimax Algorithm :

It is the strategy used by combinational search that uses heuristic to speed up the search strategy. The concept of Minimax strategy can be understood with the example of two player games, in which each player tries to predict the next move of the opponent and tries to minimize that function. Also, in order to win, the player always tries to maximize its own function based on the current situation.

Heuristic plays an important role in such kind of strategies like Minimax. Every node of the tree would have a heuristic function associated with it. Based on that heuristic, it will take the decision to make a move towards the node that would benefit them the most.

Alpha-Beta Pruning :

A major issue with Minimax algorithm is that it can explore those parts of the tree that are irrelevant, leads to the wastage of resources. Hence there must be a strategy to decide which part of the tree is relevant and which is irrelevant and leave the irrelevant part unexplored. Alpha-Beta pruning is one such kind of strategy.

The main goal of Alpha-Beta pruning algorithm is to avoid the searching those parts of the tree that do not have any solution. The main concept of Alpha-Beta pruning is to use two bounds named **Alpha**, the maximum lower bound, and **Beta**, the minimum upper bound. These two parameters are the values that restrict the set of possible solutions. It compares the value of the current node with the value of alpha and beta parameters, so that it can move to the part of the tree that has the solution and discard the rest.

Description

Alpha-Beta pruning is a optimization technique used in minimax algorithm. The idea behind this algorithm is cut off the branches of game tree which need not to be evaluated as better move exists already.

This algorithm introduces two new fields –

- Alpha – This is best value(maximum) that maximizer player can guarantee at current level or its above level
- Beta – This is the best value(minimum) that minimizer player can guarantee at the current level or its above level.

Propositional Logic :

Definition: A proposition is a statement that can be either true or false; it must be one or the other, and it cannot be both. Propositions are atomic.

An atomic proposition is one whose truth or falsity does not depend on the truth or falsity of any other proposition.

A contradiction is a statement that is always false.

A tautology is a statement that always gives a true value.

An argument is a sequence of statements.

The last statement is called the conclusion, all the previous statements are premises, or assumptions/hypotheses.

The connectives we introduce are:

- \wedge and (& or .)
- \vee or (| or +)
- \neg not (\sim)
- \Rightarrow implies (\supset or \rightarrow)
- \Leftrightarrow iff

To summarize shortly We defined 6 logical operators:

- | | |
|--|-----------------------|
| \wedge conjunction (and) | $p \wedge q$ |
| \vee disjunction (or) | $p \vee q$ |
| \neg negation (not) | $\neg p$ |
| \rightarrow conditional (if then) | $p \rightarrow q$ |
| only if | p only if q |
| \leftrightarrow biconditional (if and only if) | $p \leftrightarrow q$ |

We also saw several important logical equivalences,

RESOLUTION :

Definition: Resolution yields a complete inference algorithm when coupled with any complete search algorithm. Resolution makes use of the inference rules.

Resolution performs deductive inference. Resolution uses proof by contradiction. One can perform Resolution from a Knowledge Base. A Knowledge Base is a collection of facts or one can even call it a database with all facts.

Steps for Resolution

- Convert the given statements in Predicate/Propositional Logic
- Convert these statements into Conjunctive Normal Form
- Negate the Conclusion (Proof by Contradiction)
- Resolve using a Resolution Tree (Unification)

EXAMPLE

Now let us see an example which uses resolution. Problem Statement:

1. Ravi likes all kind of food.
2. Apples and chicken are food
3. Anything anyone eats and is not killed is food
4. Ajay eats peanuts and is still alive
5. Rita eats everything that Ajay eats Prove by resolution that Ravi likes peanuts using resolution.

Solution: Step 1: Converting the given statements into Predicate/Propositional Logic

- i. $\forall x : \text{food}(x) \rightarrow \text{likes}(\text{Ravi}, x)$
- ii. $\text{food}(\text{Apple}) \wedge \text{food}(\text{chicken})$
- iii. $\forall a : \forall b : \text{eats}(a, b) \wedge \text{killed}(a) \rightarrow \text{food}(b)$
- iv. $\text{eats}(\text{Ajay}, \text{Peanuts}) \wedge \text{alive}(\text{Ajay})$
- v. $\forall c : \text{eats}(\text{Ajay}, c) \rightarrow \text{eats}(\text{Rita}, c)$
- vi. $\forall d : \text{alive}(d) \rightarrow \sim \text{killed}(d)$
- vii. $\forall e : \sim \text{killed}(e) \rightarrow \text{alive}(e)$ Conclusion: $\text{likes}(\text{Ravi}, \text{Peanuts})$

Step 2: Convert into CNF

- i. $\sim \text{food}(x) \vee \text{likes}(\text{Ravi}, x)$
- ii. $\text{Food}(\text{apple})$
- iii. $\text{Food}(\text{chicken})$
- iv. $\sim \text{eats}(a, b) \vee \text{killed}(a) \vee \text{food}(b)$
- v. $\text{Eats}(\text{Ajay}, \text{Peanuts})$
- vi. $\text{Alive}(\text{Ajay})$
- vii. $\sim \text{eats}(\text{Ajay}, c)$
- viii. $\text{V eats}(\text{Rita}, c)$
- ix. $\sim \text{alive}(d) \vee \sim \text{killed}(d)$
- x. $\text{Killed}(e) \vee \text{alive}(e)$

Conclusion: $\text{likes}(\text{Ravi}, \text{Peanuts})$

Step 3: Negate the conclusion $\sim \text{likes}(\text{Ravi}, \text{Peanuts})$

Step 4: Resolve using a resolution tree

Important Questions from JNTUH Previous Question Papers :

- | | |
|--|---|
| 1. Define 8-Queen's problem with respect to CSP? | 3 |
| 2. What is Propositional Logic? | 3 |

- | | |
|---|----|
| 3. Explain Constraint Satisfaction Problem? | 3 |
| 4. Write Minimax Algorithm? | 5 |
| 5. What is Alpha-Beta Pruning in Search? | 4 |
| | |
| 6. Define Heuristic search? What are the advantages of Heuristic search? | 8 |
| 7. Describe the mini max algorithm with example? | 8 |
| 8. Explain the state space representation of Water-Jug Problem | 8 |
| 9. What are constraint satisfaction problem? How can you formulate them as search problem? | 10 |
| 10. Explain 8-Queens Problem along with solution with respect to CSP? | 10 |
| 11. What is FOPL? Explain the Horn clause and procedure of clause conversion with an example? | 12 |
| 12. What is Predicate logic? Explain predicate logic representation with reference to suitable example? | 12 |

Multiple Choice :

- Which of the following is one or more states that would be acceptable as solution to the problem.
A) Initial state B) Goal state C) Internal state D) All
- Which of the following that specifies the order in which the rules will be compared to the database and a way of resolving the conflicts.
A) Initial state B) Control strategy C) Internal state D) All
- Which of the following algorithm that follows backtracking while searching
A) Session Layer B) BSF algorithm C) Network Layer D) DFS algorithm
- Which of the following algorithm that follows level wise searching pattern of a tree or graph
A) Session Layer B) BSF algorithm C) Network Layer D) DFS algorithm
- Which of the following algorithm that improves the efficiency of the search process

A) Serial search B) Parallel search C) Synchronous D) Heuristic search

6. Which of the following in which solution steps can be ignored

A) Wireless B) Ignorable C) Utp D) Fiber Optics

7. Which of the following in which solution steps can be undone

A) Wireless B) Recoverable C) Utp D) Fiber Optics

8. Which of the following in which solution steps can not be undone

A) Wireless B) Irrecoverable C) Utp D) Fiber Optics

9. Which of the following is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators.

A) Simplex B) Half Duplex C) Full Duplex D) Agent

10. Which of the following that describes the agent function.

A) Agent behavior B) Eternet C) Intelligent agent D) All

UNIT -IV

Planning is the process of computing several steps of a problem-solving procedure before executing any of them. This problem can be solved by search. The main difference between search and planning is the representation of states. In search, states are represented as a single entity (which may be quite a complex object, but its internal structure is not used by the search algorithm). In planning, states have structured representations (collections of properties) which are used by the planning algorithm.

Planning systems do the following:

- 1) open up action and goal representation to allow selection
- 2) divide-and-conquer by subgoaling
- 3) relax requirement for sequential construction of solutions

Assumptions are:

- 1) Environment is deterministic
- 2) Environment is observable
- 3) Environment is static (it only in response to the agent's actions)

Planning Domain Definition Language A bit more relaxed than STRIPS. Preconditions and goals can contain negative literals.

- Action: Buy(x)
- Precondition: At(p), Sells(p,x)
- Effect: Have(x) is called an action **schema**

Planning is a key ability for intelligent systems, increasing their autonomy and flexibility through the construction of sequences of actions to achieve their goals. It has been an area of research in artificial intelligence for over three decades. Planning techniques have been applied in a variety of tasks including robotics, process planning, web-based information gathering, autonomous agents and spacecraft mission control.

Planning involves the representation of actions and world models, reasoning about the effects of actions, and techniques for efficiently searching the space of possible plans. This course will focus on the basic foundations and techniques in planning and survey a variety of planning systems and approaches.

The class will be run as a lecture course with hands-on experience with state-of-the-art planning systems. Topics covered in the course will include: action and plan representation, reactive systems, hierarchical and abstraction planning, case-based planning, machine learning in planning, multi-agent planning, interacting with the

environment, planning under uncertainty, and recent applications such as web service composition and workflow construction on the computational Grid.

Planning Algorithms

Introduce *planning as search* . There are two approaches:

- **Searching a *World Space*:**

Each node in the graph denoted a state of the world. Arcs in the graph correspond to the execution of a specific action. The planning problem is to find a path from the initial state to the goal state. There are two algorithms:

- **Progression:** An algorithm that searches for the goal state by searching through the states generated by actions that can be performed in the given state, starting from the initial state.
- **Regression:** An algorithm that searches backward from the goal state by finding actions whose effects satisfy one or more of the posted goals, and posting the chosen action's preconditions as goals (*goal regression*).

Both the algorithms are *sound* (if a plan is returned, will it work?) and *complete* (if a plan exists, does the algorithm guarantees that it will find it?). In most situations regression is a better strategy.