Artificial Intelligence
Class Assignment - 1

by Utkarsh Gupta
(A2305211551)

Q1. Discuss following applications of artificial intelligence:

(i) Games:

Game playing is a search problem Defined by
- Initial state
- Successor function
- Goal test
- Path cost / utility / payoff function

Games provide a structured task wherein success or failure can be measured with least effort. Game playing shares the property that people who do them well are considered to be displaying intelligence. There are two major components of game playing, viz. a plausible move generator, and a static valuation function generator. Plausible move generator is used to expand

orgeneratesonlyselected moves. Static evaluationfunctiongenerator,basedon heuristic generates the statievaluationfunctionvaluefor each & everymove thatisbeingmade.

Characteristics:
- "Unpredictable" opponent: solutionisa strategy specifying a move foreverypossibleopponent reply.
- Time limits:Unlikely to findgoal, must approximate.

(ii) Theorem Proving:
Theorem provinghasthe propertythatpeople who do them well are consideredto be displaying intelligence. The Logic Theoristwas an early attempt to prove mathematicaltheorems. It was abletoprove severaltheorems from the Quasells PrincipiMathematica. Gelerntera' theorem proverexploredanotherareaof mathematics: geometry. There are three types of problems in A.I. Ignorableproblems, inwhichsolutionsteps can be ignored; recoverableproblems

irrecoverable in which solution steps cannot be undone. Theorem proving falls into the first category i.e. it is ignorable suppose we are trying to solve a theorem, we proceed by first proving a lemma that we think will be useful. Eventually we realize that the lemma is not help at all. In this case we can simply ignore that lemma, and can start from beginning.

There are two basics methods of theory proving.
- Start with the given axioms, use the rules of inference and prove the theorem.
- Prove that the negation of the result cannot be TRUE.

(iii) Natural Language Processing:
The utility of computers is often limited by communication difficulties The effective use of a computer traditionally has involved the use of a programming language or a set of commands that you must use to communicate with the computer. The goal of natural language processing is to enable

people and computer to communicate in a "natural" (human) language, such as a English, rather than in a computer language.

The field of natural language processing is divided into the two sub-fields of:
- Natural language understanding, which investigates methods of allowing computer to comprehend instructio given in ordinar English so that computers can understand people more easily.
- Natural language egeneration, which strives to have computers produce ordinar English languages so that people can understand computers more easily.

(iv) Vision and Speech Processing:
The focus of natural language processing is to enable computers to communicate interactively with English words and sentences that are typed on paper or displayed on a screen. However, the primary interactiv method of communication used by humans is not reading and writing, it

isspeech.
The goal of speech processing research is to allow computers to understand human speech so that they can hear our voices and recognize the words we are speaking. Speech recognition research seeks to advance the goal of natural language processing by simplifying the process of interactive communication between people and computers. It is a simple task to attach a camera to computer so that the computer can receive visual images. It has proven to be a far more difficult task, however, to interpret those images so that the computer can understand exactly what it is seeing. People generally use vision as their primary means of sensing their environment; we generally see more than we hear, feel, smell or taste. The goal of computer vision research is to give computers this same powerful facility for understanding their surroundings. Currently, one of the primary uses of computer vision is in the area of robotics

(v) Robotics:

A robotisan electromechanical device thatcan be programmed to perform manualtasks. The Robotic Industries Association formallydefines a robotas "a reprogrammable multifunctional manipulatordesigned to move materialparts, toolsorspecializeddevices throughvariable programmed motions forthe performance of a varietyof tasks." An "intelligent" robotincludes some kindof sensoryapparatussuch as a camera, thatallowsitto respondto changes in its environment, ratherthan justto follow instructions mindlessly."

(vi) Expert Systems:

An expert system isa computerprogram designed to act as an expert in a particular domain (area of expertise). Also known as a knowledge-based system, an expert system typicallyincludesa sizable knowledge base, consistingof facts aboutthe domain and heuristic (rules) forapplyingthose facts. Expert system currentlyisdesigned to assistexperts, not to replacethem.

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

They have proven to be useful in diverse areas such as computer system configuration.
A "knowledge engineer" interviews experts in a certain domain and tries to embody their knowledge in a computer program for carrying out some task. How well this works depends on whether the intellectual mechanisms required for the task are within the present state of AI. When this turned out not to be so, there were many disappointing results One of the first expert systems was MYCIN in 1974, which diagnosed bacterial infections of the blood and suggested treatments. It did better than medical students or practicing doctors, provided its limitations were observed. Namely, its ontology included bacteria, symptoms, and treatments and did not include patients, doctors, hospitals death, recovery, and events occurring in time. Its interactions depended on a single patient being considered. Since the experts consulted by the knowledge engineers knew about patients, doctors, death, recovery etc, it is clear

that the knowledge engineers forced what the experts told them into a predetermined framework. In the present state of AI, this has to be true. The usefulness of current expert systems depends on their users having common sense.

Q2. Discuss various types of production systems and their suitability different AI problems.

A production system in AI helps create AI-based computer programs. With the help of it, the automation of various types of machines has become an easy task. The types of machines can be a computer, mobile applications, manufacturing tools, or more. The set of rules in a production system defines the behavior of the machine. It helps the machine respond to the surroundings.

A monotonic production system
A non-monotonic production system
A partially commutative production system
A commutative production system.

Monotonic Production System:
It's a production system in which the application of a rule never prevents the later application of another rule, that could have also been applied at the time the first rule was selected.

Partially Commutative Production System: It's a type of production system in which the application of a sequence of rules transforms state X into state Y, then any permutation of those rules that is allowable also transforms state x into state Y. Theorem proving falls under the monotonic partially commutative system. Blocks world and 8 puzzle problems like chemical analysis and synthesis come under monotonic, not partially commutative systems.

Although playing the game of bridge comes under non-monotonic, not partially commutative system. For any problem, several production systems do exist. Some will be efficient than others.

Non-Monotonic Production Systems are useful for solving ignoral-l problems. These systems are important for man implementation standpoint because they can be implemented without the abilit to go back track back to previous states when it is discovered that atan incorrect path was followed. This production system increases the efficiency since it is not necessary to keep track of the changes made in the search process.

Commutative Systems are usually useful for problems in which changes occur but can be reversed and in which the order of operation is not critica for example the, 8 puzzle problem. Production systems that are not usually got partially commutative are useful for many problems in which irreversibl changes occur, such as chemical analysis. When dealing with such systems, the order in which operations are performed is very important and hence correct decisions must be made at the first time itself.

Q3. Consider a water jug problem. You are given two jugs a 4-gallon one and a 3-gallon one, a pump which has unlimited water which you can use to fill the jug and the ground on which water may be poured. Neither jug has any measuring markings on it. How can you get exactly 2 gallons of water in the 4-gallon jug?
State the production rules for the water jug problem.

State Representation and Initial State:
we will represent a state of the problem as a tuple (x, y) where x represents the amount of water in the 4-gallon jug and y represents the amount of water in the 3-gallon jug.
Note $0 < x < 4$ and $0 < y < 3$.

Our initial state: (0, 0)

Goal Predicate:
state = (2, y) where $0 < y < 3$

Operators:
we must define a set of operators that will

take us from one state to another:

1. Fill 4-gal jug (x, y) -> (4, y)
   x < 4

2. Fill 3-gal jug (x, y) -> (x, 3)
   y < 3

3. Empty 4-gal jug on ground
   (x, y) -> (0, y)
   x > 0

4. Empty 3-gal jug on ground
   (x, y) -> (x, 0)
   y > 0

5. Pour water from 3-gal jug to fill 4-gal
   (x, y) -> (4, y-(4-x))
   0 < x+y >= 4 & y > 0

6. Pour water from 4-gal jug to fill 3-gal
   (x, y) -> (x-(3-y), 3)
   0 < x+y >= 3 & x > 0

1. Pour all of water from 3-gal to 4-gal jug

$(x, y) \rightarrow (x+y, 0)$

$0 < x+y <= 4 \ \& \ y >= 0$

8. Pour all of water from 4-gal to 3-gal jug

$(x, y) \rightarrow (0, x+y)$

$0 < x+y <= 3 \ \& \ x >= 0$

Through Graph Search, the following solutions is found :

Gals in 4-gal	Gals in 3-gal	Rule Applied
0	0	1
4	0	6
1	3	4
1	0	8
0	1	1
4	1	6
2	3	

Q4. Explain the algorithm of Best First Search (BFS). Give an example where BFS would work better than breadth first search and depth-first search.

In BFS and DFS, when we are at a node, we can consider any of the adjacent as next node. So both BFS and DFS blindly explore paths without considering any cost function. The idea of Best First Search is to use an evaluation function to decide which adjacent is most promising and then explore. Best First Search falls under the category of Heuristic Search or Informed Search

Algorithm:
- Create 2 empty lists OPEN and CLOSED
- Start from the initial node (say N) and put it in the 'ordered' OPEN list
- Repeat the next steps until GOAL node is reached
- If the OPEN list is empty, then EXIT the loop returning 'False'
- Select the first/top node (say N) in the OPEN list and move it to the CLOSED list.

Also, capture the information of the parent node
- If N is a GOAL node, then move the node to the Closed list and exit the loop returning 'True'. The solution can be found by backtracking the path
- If N is not the GOAL node, expand node N to generate the 'immediate' next nodes linked to node N and add all those to the OPEN list
- Reorder the nodes in the OPEN list in ascending order according to an evaluation function f(n).

This algorithm will traverse the shortest path first in the queue. The time complexity of the algorithm is given by O(n*log n).

Q5. Solve the following CryptArithmetic problem.

```
  DONALD
+ GERALD
----------
  ROBERT
```

DONALD + GERALD = ROBERT

Given 'D=5'

(If not given assume D=5 at the initial stage)

6 5 4 3 2 1

D O N A L D

+ G E R A L D

c1 c2 c3 c4 c5

R O B E R T

1. 'D=5' is assumed, so 'D+D=T' therefore 'T=0' & 'c5=1'.

2. In column 5 'O+E=O' as 'T=0' so E cannot be 0, therefore 'E=9'.
'O+9=O' is possible if 'c2=1'. Therefore 'c2=1'.

3. In column 3 'A+A=9', but the addition of any 2 same number is always even, given that addition is 9 which is possible when there is carry therefore 'c4=1', so 'A=4'.

4. Remaining numbers to be assigned are {1,2,3,6,7,8} to

{O,N,R,B,L,G}.

5. We have 'E=9' & 'c2=1' so from column 5 we get 'c1=1'. Also from column 2 we have 'L+L+c5=R' where 'c5=1' therefore R is odd so R can be 1 or 3 or 7. As 'D+G' does not generate carry shown in column 6 so R cannot be 1 or 3. Therefore 'R=7' & 'G=1'.

6. We have 'R=7' so from column 2 we have 'L+L+1=17', therefore 'L=8'.

7. From column 3 we get that 'A+A+c4=E' and so there is no carry, therefore 'c3=0'.

8. From column 4 we get 'N+R+c3=B' we have R=7 & 'c3=0', so 'N+7=10+B', therefore 'N=B+3'. {2,3,6} are remaining to be assigned so to satisfy the constraint 'N=B+3' we get 'B=3' & 'N=6'.

9. And remaining 'O=2'.

5 2 6 4 8 5

+ 1 9 7 4 8

---

1 2 3 9 1 0

VALUES:

D=5

O=2

N=6

A=4

L=8

G=1

E=9

R=1

B=3

T=0

Q1. Discuss the A* algorithm Compare it with a hill-climbing search method.

A* Search algorithm is one of the best and popular techniques used in path finding and graph traversal. A* Search algorithms unlike other traversal techniques, it has "brains".

What it means is that it is really a smart algorithm that separates it from the other conventional algorithms. This fact is cleared in detail in the below sections.
And it is also worth mentioning that many games and web-based maps use this algorithm to find the shortest path very efficiently (approximation).

A* algorithm has 3 parameters:

g : the cost of moving from the initial cell to the current cell. Basically, it is the sum of all the cells that have been visited since leaving the first cell.

h : also known as the heuristic value. it is the estimated cost of moving from the current cell to the final cell. The actual cost cannot be calculated until the final cell is reached. Hence, h is the estimated cost. We must make sure that there is never an overestimation of the cost.

f: it is the sum of g and h

So, $f = g + h$

The way that the algorithm makes its decisions is by taking the f-value into account. The algorithm selects the smallest f-valued cell and moves to that cell. This process continues until the algorithm reaches its goal cell.

Algorithms like weighted A* systematically explore the search space in 'best' first order. 'Best' is defined by a node ranking function which typically considers the cost of arriving at a node, g, as well as the estimated cost of reaching a goal from a node, h.
Some algorithms such as A* also consider the distance of a node from the goal. Hill climbing algorithms are less deliberative, rather than considering all open nodes, they expand the most promising descendant of the most recently expanded node until they encounter a solution.

Q9. Find the path to reach from S to G using A* search.

Starting from S, the algorithm computes g(x) + h(x) for all nodes in the fringe at each step, choosing the node with the lowest sum. The entire work is shown in the table below.

Note that in the fourth set of iteration we get two paths with equal summed cost f(x), so we expand them both in the next set. The path with lower cost on further expansion is the chosen path.

Path	h(x)	g(x)	f(x)
S	↑	0	↑

S -> A 9 3 12

S -> D == 5 2 ↑

S -> D -> B == 4 2 + 1 = 3 ↑

S -> D -> E = 13 2 + 4 = 6 9

S -> D -> B -> C == 2 3 + 2 = 5 ↑

S -> D -> B -> E == 3 3 + 1 = 4 ↑

S -> D -> B -> C -> G != 0 5 + 4 = 9 9

S -> D -> B -> E -> G == 0 4 + 3 = 7 7

Path: S -> D -> B -> E -> G

Cost: 7.

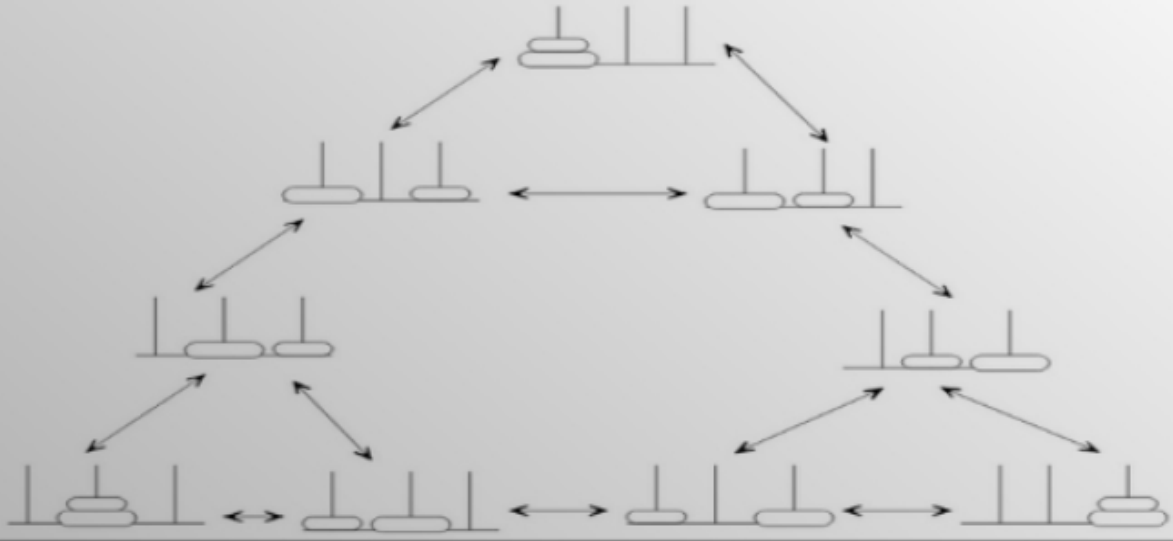
Q10. Before we can solve a problem using a state-space search, we must define an appropriate state space. For each of the problems

(a) Tower of Hanoi

(b) Monkeys and bananas, find a good state-space representation.

Towers Hanoi

A possible state-space representation of the Towers Hanoi problem using a graph is indicated in the below figure. The legal moves in this state-space involve moving one ring from one pole to another, moving one ring at a time, and ensuring that a larger ring is not placed on a smaller ring.



a) The Monkey & Bananas Problem

A monkey is in a cage and bananas are suspended from the ceiling, the monkey wants to eat a banana but cannot reach then in the room is a chair and a stick if the monkey stands on the chair and waves the stick, he can knock a banana down to eat it
what are the actions the monkey should take?

Initia state:
monkey on ground
with empty hand
bananas suspended

Goal state:

monkey eating

Actions:

climb-chair get off

grab X

wave X

eat X