National University of Computer and Emerging Sciences
FAST School of Computing Spring-2023 Islamabad Campus

MT 2002: Artificial Intelligence

Serial No:

Sessional Exam-I Solution

Total Time: 1 Hour
Total Marks: 70

Monday, 26 Sep, 2023

Course Instructors

Dr. Shahnawaz Qureshi

____
Signature of Invigilator

Student Name ____________ Roll No. ____________ Course Section ____________ Student Signature ____________

DO NOT OPEN THE QUESTION BOOK OR START UNTIL INSTRUCTED.

Instructions:

1. Attempt on question paper. Attempt all of them. Read the question carefully, understand the question, and then attempt it.
2. No additional sheet will be provided for rough work. Use the back of the last page for rough work.
3. If you need more space, write on the back side of the paper and clearly mark question and part number etc.
4. After asked to commence the exam, please verify that you have **nine (9)** different printed pages including this title page. There are a total of **4** questions.
5. Calculator sharing is strictly prohibited.
6. Use permanent ink pens only. Any part done using soft pencil will not be marked and cannot be claimed for rechecking.

	Q-1	Q-2	Q-3	Q-4	Total
Marks Obtained					
Total Marks	20	10	20	20	70

Question No. 1 Agents [16+4+4+4+2=30]

(a) For each of the following terms on the left, write in the letter corresponding to the best answer or the correct definition on the right. The first one is done for you as an example.

A	Agent	A	Perceives environment by sensors, acts by actuators
K	Percept	B	All states reachable from the initial state by a sequence of actions
P	Performance Measure	C	Guaranteed to find a solution if one is accessible
L	Rational Agent	D	Process of removing detail from a representation
B	State Space	E	Maximum number of successors of any node
I	Search Node	F	Set of all leaf nodes available for expansion at any given time
N	Link between nodes	G	Estimates cost of cheapest path from current state to goal state
J	Path	H	Guaranteed to find lowest cost among all accessible solutions
D	Abstraction	I	Represents a state in the state space
H	Optimal Search	J	Sequence of states connected by a sequence of actions
C	Complete Search	K	Agent's perceptual inputs at any given instant
M	Expand a state	L	Agent that acts to maximize its expected performance measure
F	Frontier	M	Apply each legal action to a state, generating a new set of states
O	Search Strategy	N	Represents an action in the state space
E	Branching Factor	O	How a search algorithm chooses which node to expand next
G	Heuristic Function	P	Evaluates any given sequence of environment states for utility

(b) A **task environment** is defined as a set of four items, with acronym PEAS. Fill in the blanks with the names of the PEAS components. **(4)**

P___Performance_____________________, **E**_____Environment_____________________________

A_________Actutaors____________________, **S**______Sensors____________________________

(c) A **learning agent** consists of four components, which are, **(4)**

(1)_______ Sensors: ___________, **(2)**______ Perception/Cognition: _________
(3)_ Learning/Reasoning ___________________________, **(4)**____Actuators ____

(d) List some environmental properties that are important for intelligent agents. Describe the properties of the environment for the Mars Rover. (4)

Fully/partially observable, discrete/continuous, single/multi agent, stochastic/deterministic, adversarial, dynamic/static. Mars rover: partially observable (because sensors are limited), single agent, continuous, stochastic (e.g., weather events), dynamic (slips?), non-adversarial.

(e) Are reflex actions (such as flinching from a hot stove) rational? Are they intelligent? (2)

Yes, they are rational, because slower, deliberative actions would tend to result in more damage to the hand. If "intelligent" means "applying knowledge" or "using thought and reasoning" then it does not require intelligence to make a reflex action.

Question No. 2 State Space Search (Problem Formulation) [4+4+2+2=12]

Suppose two friends live in different cities on a map, such as the Romania map. On every turn, we can simultaneously move each friend to a neighboring city on the map. The amount of time needed to move from city *i* to neighbor *j* is equal to the road distance *d(i, j)* between the cities, but on each turn the friend that arrives first must wait until the other one arrives (and calls the first on his/her cell phone) before the next turn can begin. We want the two friends to meet as quickly as possible.

a. Write a detailed formulation for this search problem. **(4)**

State space: States are all possible city pairs (i,j). The map is not the state space.

Successor function: The successors of (i,j) are all pairs (x,y)such that Adjacent(x,i) and Adjacent(y,j).

Goal: Be at (i,i) for some i. Step cost function:

The cost to go from (i,j) to (x,y) is max(d(i,x),d(j,y)).

b. Let *D(i, j)* be the straight-line distance between cities *i* and *j*. Which of the following heuristic functions are admissible? (i) *D(i, j);* (ii) 2. *D(i, j);* (iii) *D(i, j)/2*. **(4)**

Hence (iii) is admissible.

c. Are there completely connected maps for which no solution exists? **(2)**

Yes

d. Are there maps in which all solutions require one friend to visit the same city twice? **(2)**

Yes

Question No. 3 Uninformed Search [2+2+2+2+2=10]

Which of the following are true and which are false? Explain your answers.

a. Depth-first search always expands at least as many nodes as A∗ search with an admissible heuristic.

False

b. b. h(n)=0 is an admissible heuristic for the 8-puzzle.

True

c. A∗ is of no use in robotics because percepts, states, and actions are continuous.

True

d. Breadth-first search is complete even if zero step costs are allowed.

True

e. Assume that a rook can move on a chessboard any number of squares in a straight line, vertically or horizontally, but cannot jump over other pieces. Manhattan distance is an admissible heuristic for the problem of moving the rook from square A to square B in the smallest number of moves.

False

Question No. 4 Uninformed Search [5+5+5=15]

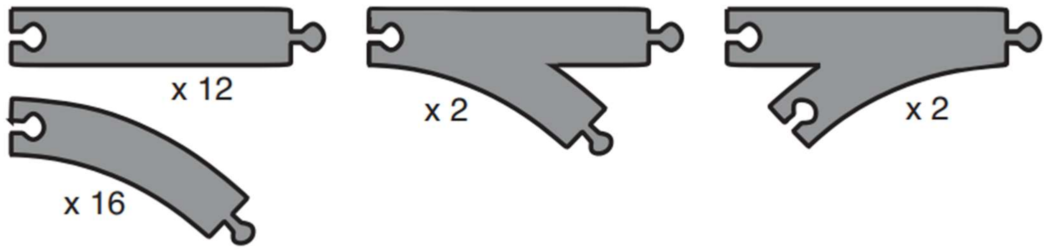

Figure 3.32 The track pieces in a wooden railway set; each is labeled with the number of copies in the set. Note that curved pieces and “fork” pieces (“switches” or “points”) can be flipped over so they can curve in either direction. Each curve subtends 45 degrees.

A basic wooden railway set contains the pieces shown in Figure 3.32. The task is to connect these pieces into a railway that has no overlapping tracks and no loose ends where a train could run off onto the floor.

a. Suppose that the pieces fit together exactly with no slack. Give a precise formulation of the task as a search problem.

Initial state: one arbitrarily selected piece (say a straight piece).

Successor function: for any open peg, add any piece type from remaining types. (You can add to open holes as well, but that isn’t necessary as all complete tracks can be made by adding to pegs.) For a curved piece, add in either orientation; for a fork, add in either orientation and (if there are two holes) connecting at either hole. It’s a good idea to disallow any overlapping configuration, as this terminates hopeless configurations early. (Note: there is no need to consider open holes, because in any solution these will be filled by pieces added to open pegs.)

Goal test: all pieces used in a single connected track, no open pegs or holes, no overlapping tracks. Step cost: one per piece (actually, doesn’t really matter).

b. Identify a suitable uninformed search algorithm for this task and explain your choice.

All solutions are at the same depth, so depth-first search would be appropriate. (One could also use depth-limited search with limit $n-1$, but strictly speaking it’s not necessary to do the work of checking the limit because states at depth $n-1$ have no successors.) The space is very large, so uniform-cost and breadth-first would fail, and iterative deepening simply does unnecessary extra work. There are many repeated states, so it might be good to use a closed list.

c. Explain why removing any one of the “fork” pieces makes the problem unsolvable.

A solution has no open pegs or holes, so every peg is in a hole, so there must be equal numbers of pegs and holes. Removing a fork violates this property. There are two other "proofs" that are acceptable: 1) a similar argument to the effect that there must be an even number of "ends"; 2) each fork creates two tracks, and only a fork can rejoin those tracks into one, so if a fork is missing it won't work. The argument using pegs and holes is actually more general, because it also applies to the case of a three-way fork that has one hole and three pegs or one peg and three holes. The "ends" argument fails here, as does the fork/rejoin argument (which is a bit handwavy anyway).