Practice exam quiz (long answer questions)

Started: Jun 24 at 22:31

Quiz Instructions

These are practice 'long answer' questions, just based on the sample exam questions compiled from previous years.

The subject staff have worked hard on the real exam to try to make answering questions as straightforward as possible, so hopefully it will be smoother than these sample questions, but these questions are illustrative of the type of long answer questions on the final exam.

Questions
Spacer
Question 1
Question 2
Spacer
Question 3
Question 4
Question 5
Question 6
Question 7
Spacer
Question 8

Time Elapsed: Hide
3 Days, 1 Hour, 51 Minutes, 43 Seconds

Blind and heuristic search

Question 1

0 pts

The following search algorithms can be implemented in a similar manner, differing only in the abstract data structure used to implement the open list.

- Breadth First Search
- Depth First Search
- Uniform Cost Search

Answer the following questions:

I. State which abstract data structure you would use for Breadth First Search. Briefly explain, in one or two sentences, why you would choose the particular data structure.
II. State which abstract data structure you would use for Depth First Search. Briefly explain, in one or two sentences, why you would choose the particular data structure.
III. State which abstract data structure you would use for Uniform Cost Search. Briefly explain, in one or two sentences, why you would choose the particular data structure.

12pt Paragraph

I. Breadth First Search: Queue

Breadth first search involves exploring closer nodes before further nodes, in terms of number of steps. Hence, nodes are expanded in the same order in which they are created. We use a queue (FIFO data structure) for the open list.

II. Depth First Search: Stack

Depth first search involves following a path all the way down until there are no further actions available (then backtracking). Hence, nodes are expanded in reverse order of creation (i.e. latest created node is expanded first. We use a stack (LIFO data structure) for the open list.

III. Uniform Cost Search: Priority Queue

Uniform cost search takes into account the action costs and greedily explores the best action first, based upon the cumulative action cost (sum of action costs from initial state to current state). We use a priority queue to extract the node with lowest cumulative action cost.

p 149 words

Question 2

0 pts

Recall that A' requires a consistent heuristic to guarantee finding an optimal plan without reopening nodes. Draw or define a graph and make up an admissible but inconsistent heuristic function where A' returns a suboptimal solution.

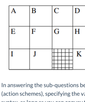
Write down the steps involved in the A' algorithm by devising a simple example to illustrate.

Note: avoid making large examples, a graph with 4 nodes should be sufficient.

12pt Paragraph

p 0 words

The next five questions refer to the following example. They concern a classical planning problem where a robot can move horizontally and vertically to adjacent cells as depicted in the figure below. Note that the robot cannot move diagonally between cells and the hashed cell is inaccessible to the robot.



In answering the sub-questions below, you are allowed to use variables as arguments for the actions (action schemes), specifying the values of the variables. Note: it is **not** compulsory to use PDDL syntax, as long as you can convey the main ideas.

Hint: consider that the position of the robot could be modelled as either:

- row/column tuples, e.g. h0, 0i could refer to the lower left cell, or
- single cells, e.g. position hii could refer to the lower left cell.

Question 3

0 pts

Describe briefly in STRIPS how to model the problem where a robot can move horizontally and vertically among adjacent cells, such that the h_{man} heuristic estimates the same values as the Manhattan heuristic

12pt Paragraph

Atoms

F = {at(x), adjacent(x, x'), wall(x) | x ∈ {A, B, ..., K}}

Operations

O = {move(x, x')}

Pre(move(x, x')) = {at(x), ¬wall(x'), adjacent(x, x')}
Add(move(x, x')) = {at(x')}
Del(move(x, x')) = {at(x)}

In this model, once at a position, the agent will always be at that position in the delete relaxation problem.

p 67 words

Question 4

0 pts

Describe briefly in STRIPS how to model the problem where a robot can move horizontally and vertically among adjacent cells, such that the h_{man} heuristic estimates the same values as the Optimal heuristic.

12pt Paragraph

Atoms

F = {row(r), col(c), wall(r, c) | r, c ∈ {0, 1, 2}, y ∈ {0, 1, 2, 3}}

Operations

O = {move(r, c, r', c')}

Pre(move(r, c, r', c')) = {row(r), col(c), ¬wall(row, col), ((r' - r| = 1 ∧ c = c') ∨ (|c' - c| = 1 ∧ r = r')}
Add(move(r, c, r', c')) = {row(r'), col(c')}
Del(move(r, c, r', c')) = {row(r), col(c)}

In this model, once in a **row**, the agent will always be in that **row** and once in a **column**, the agent will always be in that **column**.

p 110 words

Question 5

0 pts

Using your last STRIPS encoding where $h_{man} = h'$; an initial state s_0 = *robot at location I* and a goal state s_g = *robot at location K*, compute $h_{man}(s_0)$ and h_F from the best supporters induced by h_{add}.

Show your working

12pt Paragraph

p 0 words

Question 6

0 pts

We have 2 rooms A and B, 2 objects o1 and o2. A robot can load and unload objects if the robot is at the same location, and move from one room to the other. We want to get o1 and o2 into room B, given that both are initially at A.

Model the problem in STRIPS in such a way that the optimal plan would be the following:

```
pick(o1, A), move(A, B), drop(o1, B), move(B, A), pick(o2, A), move(A, B), drop(o2, B)
```

12pt Paragraph

Atoms

F = {atl(obj, room), adj(room, room'), robotAt(room), holding(obj), armFree
{obj ∈ {o1, o2}, room ∈ {A, B}}

Operations

O = {pick(obj, room), move(room, room'), drop(obj, room)}

Pre(pick(obj, room)) = {robotAt(room), atl(obj, room), armFree}
Add(pick(obj, room)) = {holding(obj)}
Del(pick(obj, room)) = {atl(obj, room), armFree}

Pre(move(room, room')) = {robotAt(room), adj(room, room')}
Add(move(room, room')) = {robotAt(room')}
Del(move(room, room')) = {robotAt(room)}

Pre(drop(obj, room)) = {robotAt(room), holding(obj)}
Add(drop(obj, room)) = {atl(obj, room), armFree}
Del(drop(obj, room)) = {holding(obj)}

Initial situation

I = {atl(o1, A), atl(o2, A), adj(A, B), adj(B, A), robotAt(A), armFree}

Goals

G = {atl(o1, B), atl(o2, B)}

p 137 words

Question 7

0 pts

What's the $h_{max}(s_0)$ value of your STRIPS model, where s_0 stands for the initial state?

Show your working

12pt Paragraph

p 0 words

MDPs and reinforcement learning

The next six questions refer to the following example

Imagine a kitchen robot whose task is to ask the user what kinds of cereals he/she wants for breakfast, wait for the user answer, and then hand out the appropriate cereal box once it knows the desired cereal. We want to design a simple dialogue system to handle the interaction with the user.

A simple way to model it is via a discount-reward MDP with only two states: *UnknownCereal*, where the robot does not know which cereal to give, and *KnownCereal*, where the robot knows the cereal to hand out.

There are only two possible actions in the model:

- Action *AskCerealType* corresponds to the robot asking the user for the cereal box he/she wishes to have. The action is only available in state *UnknownCereal*, and has a reward of -1 (a cost) in that state.
  The probability of reaching state *KnownCereal* is 0.8 (if the user answers the robot's question), and otherwise (probability 0.2) the robot remains in *UnknownCereal* (if the user ignores the question or provides an unclear answer).
- Action *GiveCereal* corresponds to the robot physically giving the cereal to the user. The action is available in the state *KnownCereal*, and has a reward of 5.
  When the robot executes this action in state *KnownCereal*, the MDP reaches an absorbing goal state and finishes. As such, there is no reward available from this state.
  When the robot executes this action in the *UnknownCereal* state, the MDP reaches the absorbing goal state with probability 0.3 (it gets lucky and hands the right cereal) and receives a reward of 5, or the person rejects the cereal and the robot goes back to the *UnknownCereal* state with probability 0.7 and receives a reward of -2.

Question 8

0 pts

Assuming a discount factor γ = 0.9, calculate the the value function *V* for each of the states *UnknownCereal* and *KnownCereal* using value iteration, for the 2nd and 3rd iterations.

Show your working.

Iteration 1:
V0(KnownCereal) = 0.0
V0,UnknownCereal) = 0.0

12pt Paragraph

Iteration 2

V1(KnownCereal)

= max(Q(UnknownCereal, AskCerealType), Q(UnknownCereal, GiveCereal))
= max((0.2(-1 + 0.9(0.0)) + 0.8(-1 + 0.9(0.0)), 0.7(-2 + 0.9(0.0)) + 0.3(5 + 0.9(0.0)))
= max(-1, 0.1)
= 0.1

V(KnownCereal)
= max(Q(KnownCereal, GiveCereal))
= max((1.0(5 + 0.9(0)))
= max((5))
= 5

Iteration 3

V(UnknownCereal)
= max(Q(UnknownCereal, AskCerealType), Q(UnknownCereal, GiveCereal))
= max((0.2(-1 + 0.9(0.1)) + 0.8(-1 + 0.9(5)), 0.7(-2 + 0.9(0.1)) + 0.3(5 + 0.9(0)))
= max((2.618, 0.163))
= 2.618

V(KnownCereal) = max(Q(KnownCereal, GiveCereal))
= max((1.0(5 + 0.9(0)))
= max((5))
= 5

p 122 words

Question 9

0 pts

Given the value function that you calculate above, what policy would maximise the robot's expected reward?

Show your working.

12pt Paragraph

π(UnknownCereal) = $\text{argmax}_{a \in \{AskCerealType,\ GiveCereal\}}$(Q(UnknownCereal, a))
= AskCerealType

π(KnownCereal) = $\text{argmax}_{a \in \{GiveCereal\}}$(Q(KnownCereal, a))
= GiveCereal

p 19 words

Question 10

0 pts

In your own words, explain the difference between value iteration and policy iteration.

12pt Paragraph

In value iteration, we start with arbitrary state values. Every iteration, we update as follows:

For each state s:

- Set V(s) = max Q(s, a) over all possible actions a
- Set π(s) = a (where a is the action associated with the maximum above)

We do this until the maximum difference is below some threshold epsilon.

However, in **policy iteration**, we start with an arbitrary policy (not state values). Every iteration, we update as follows:

For each state s:

- Evaluate the policy (calculate V(s) following the policy)
- Update the policy with new actions

We do this until the policy converges.

Hence, value iteration updates and keeps track of state values while policy iteration updates and keeps track of a policy. Policy iteration is guaranteed to converge after a finite number of iterations, since there are only a finite number of possible policies (if state space and action space are finite).

p 153 words

Question 11

0 pts

The robot designers have found that the probabilities used for outcomes are incorrect and different for each household. As such, they
decide to instead use reinforcement learning to learn the policy after deployment.

A few weeks after deployment, one such robot has the following Q-table:

State	AskCereal	GiveSerial
UnknownCereal	7.2	1.9
KnownCereal	--	3.4

Assuming a discount reward factor of 0.9 and a learning rate of 0.5.

In the state *UnknownCereal*, the robot executes *GiveCereal*, which is rejected. It decides to execute *GiveCereal* again.

Update the Q-table both the 1-step Q-learning and 1-step SARSA updates for the first *GiveCereal* action. That is, calculate two new Q-tables.

Show your working.

12pt Paragraph

1-step Q-learning

Q(UnknownCereal, GiveCereal)
= Q(UnknownCereal, GiveCereal) + α(r + γ max_a(Q(UnknownCereal, a)) - Q(UnknownCereal, GiveCereal)
= 1.9 + 0.5(-2 + 0.9(7.2) - 1.9)
= 3.19

1-step SARSA

Q(UnknownCereal, GiveCereal)
= Q(UnknownCereal, GiveCereal) + α(r + γ Q(UnknownCereal, GiveCereal) - Q(UnknownCereal, GiveCereal)
= 1.9 + 0.5(-2 + 0.9(1.9) - 1.9)
= 0.805

p 61 words

Question 12

0 pts

Challenge Question: In the state *UnknownCereal*, the robot executes *GiveCereal* three times in a row. Each time the cereal is rejected.

Calculate the 2-step SARSA update for this execution of three actions

Assume the same parameters (discounter reward factor, etc) as in the previous question

Show your working

12pt Paragraph

The discounted reward is

G = -2 + 0.9(-2) + 0.9^2(1.9) = -2.261

The updates Q-value is

Q(UnknownCereal, GiveCereal)
= Q(UnknownCereal, GiveCereal) + α(G - Q(UnknownCereal, GiveCereal))
= 1.9 + 0.5(-2.261 - 1.9)
= -0.1805

p strong 39 words

Question 13

0 pts

The robot designers are finding that the owners of the robots are demanding a refund because the robots get their cereal preference wrong too often. Give one technique that the robot designers could do to improve the situation.

12pt Paragraph

We have an intuitiuon that GiveCereal is a bad action in the UnknownCereal state. We want to encourage the AskCereal action more. Hence, we can use **reward shaping** to encourage moving to the KnownCereal state.

Define a potential function as below:

Φ(UnknownCereal) = 0

Φ(KnownCereal) = 2

Now, when the robot performs AskCereal and ends up in state KnownCereal, a small reward is gained (rather than just incurring a cost of -1 as before). This encourages the robot to perform AskCereal in state UnknownCereal.

p 84 words

Question 14

0 pts

Imagine a reinforcement learning algorithm that monitors heart beat information from a fitness device, such as a FitBit, to determine whether a person develops a heart problem, such as an irregular heartbeat or a faster heartbeat.

List one potential ethical dilemma that could occur in such a situation. Justify why you believe this could be a serious problem.

12pt Paragraph

p 0 words

Game theory

Question 15

0 pts

Consider the following abstract two-player game in normal form. Find all pure and mixed-strategy equilibria for this game.

		Player 2		
		L	M	R
Player 1	U	2,4	3,0	1,1
	D	3,2	10,3	0,4

HINT: Consider the notion of *dominated strategies*, in which some strategies are strictly dominated by others, so can be discarded.

Show your working

12pt Paragraph

There are no pure Nash equilibria in this game. Take (U, L) with payoff (2, 4): Player 1 would rather play strategy D when Player 2 plays L. This is similar in for all other cells.

Now, consider strategy M for Player 2. This is dominated by strategy R; for all possible strategies of Player 1, the payoff of strategy R is at least as good as strategy M. Hence, strategy M can be eliminated from the game. The result is below:

	L	R
U	2, 4	1, 1
D	3, 2	0, 4

We find the mixed strategy for Player 1:

Let p = probability that Player 1 plays U

E_1(L) = 4p + 2(1-p) = 2p + 2
E_1(R) = 1p + 4(1-p) = 4 - 3p

E_1(L) = E_1(R)
2p + 2 = 4 - 3p
5p = 2
p = 2/5

We find the mixed strategy for Player 2:

Let q = probability that Player 2 plays L

E_2(U) = 2q + 1(1-q) = 1 - q
E_2(D) = 3q + 0(1-q) = 3q

E_2(U) = E_2(D)
1 - q = 3q
4q = 1
q = 1/4

Hence, the mixed strategy equilibrium is:

{2/5 U, 1/4 L}

p 214 words

Question 16

0 pts

Challenge Question: Two players, A and B play the following game.

First A must choose IN or OUT.

If A chooses OUT the game ends, and the payoffs: are A gets 2, and B gets 0.

If A chooses IN then B observes this and must then choose IN or OUT.

If B chooses OUT the game ends, and the payoffs are: B gets 2, and A gets 0.

If B chooses IN then they play the following simultaneous move game:

		Player B	
		L	R
Player A	U	3,1	0,2
	D	-1,2	1,3

Draw the extended-form tree for this game and calculate the equilibria of the extended-form game.

Note: to 'draw' the tree, you can use text format; e.g.

U -1.a -> X
X -2.b -> Y (0,1)
X -2.c -> Z (1,0)

represents a tree with root node U. From U, player 1 must select move a, and the player 2 chooses between moves b and c.

Alternatively, you may draw it by hand on paper or using a stylus, and then insert the image using

the icon at the top of the text box

12pt Paragraph

First, consider the normal form game. (D, R) is the Nash equilibrium with payoff (1, 3).

The extended-form tree is given below:

By backward induction (annotated on tree), it is apparent that the equilibrium of the extended form game is (OUT, :) with payoff (2, 0).

p

45 words

Question 17

0 pts

Challenge Question. *The following question is under-specified, but is intended to improve your understanding of subject content and general problem-solving ability, rather than act as an entirely accurate reflection of an exam question.*

Consider a person who is mugged by someone on the street with a gun. The person has an unloaded gun in their own pocket. They could get the gun out to try to scare off the mugger, however, they risk the mugger shooting them instead. If they do not get out the unloaded gun, their mugger has time to search only their left or right pocket, but not both and the mugger will not shoot them. The person has $100 in their left pocket and nothing in their right pocket, and the mugger knows this. Assume that if the mugger does not get the $100, they get no payoff.

Should the person draw their unloaded gun or not?

Justify your answer

12pt Paragraph

This can be modelled as an extended-form game. The tree is shown below:

Through backward induction (annotated on tree), it can be seen that the equilibrium is (not draw, robb). If the victim drew, the mugger would shoot and kill the victim, as this has a payoff of 100 (the mugger can easily take the $100 off the dead victim); the payoff is -∞ for the victim (dead). The victim would rather have a payoff of -100 and hence should **not draw**.

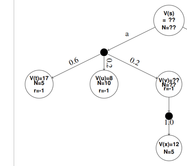
p

82 words

Question 18

0 pts

Below is a tree from MCTS in which a collection of roll-outs have been performed, and six nodes expanded are shown (others are omitted as they are not relevant to the question). The notation *V(s) = x* indicates that *v* is the value of state *s*, *N = n* indicates that the node has been visited *n* times, and *r = R* indicates that the reward *R* is given when this state is visited.



Using back-propagation, calculate the values for the states s and v.

Show your working

12pt Paragraph

Let γ = 0.9

V(v) = 1.0(0 + 0.9(12)) = **10.8**

V(s) = max{ 0.6(-1 + 0.9(17)) + 0.2(-1 + 0.9(8)) + 0.2(-1 + 0.9(10.8)), 1.0(0 + 0.9(190)) }
= max{ 11.564, 171 }
= **171**

p • strong

45 words

Question 19

0 pts

Given your answer for the previous question, what would action should be chosen at state s to maximise expected reward?

Show your working.

12pt Paragraph

Action b should be chosen. This is since Q(s, a) = 10.8 and Q(s, b) = 171 from above.

p

21 words

Saved at 0:22

Submit Quiz