

COMP90054: AI Planning for Autonomy

Revision exercises — Semester 2, 2016

Important information

The final exam will count for 50% of your final grade and will consist of 50 marks.

The final exam will be a two-hour exam, with an additional 15 minutes of reading time at the start.

You should answer/attempt all questions. There will be space in which to write answers, and only answers in that space will be marked. If you require notes for rough working, you should use the blank packages (on the reverse side of the questions).

Revision exercises

These revision exercises add up to *more than* 50 marks. They are just a collection of sample questions that you can use for practice, and are not intended to represent an exam.

The sample questions in this document are intended to reflect merely the style and difficulty of questions that you may encounter in the exam. They are in no way meant to reflect the content of the final exam. As such, it is highly recommended that you do *not* focus your study on these questions.

As with an actual exam, there will be no sample solutions provided. You are all encouraged to submit sample answers to the subject discussion board, and to provide feedback on other people's answers. The subject staff will provide input if they feel that it would be valuable.

Question 1

(20 Marks)


- (a). 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
- (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.
- (b). Recall that A* requires a consistent heuristic to guarantee finding an optimal plan without re-opening 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.

Question 2

(20 Marks)

- (a). This question concerns 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.

| | | | |
|---|---|---|---|
| A | B | C | D |
| E | F | G | H |
| I | J |  | K |

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. $\langle 0, 0 \rangle$ could refer to the lower left cell, or
 - single cells, e.g. position $\langle I \rangle$ could refer to the lower left cell.
- (i). Describe briefly in STRIPS how to model the problem where a robot can move horizontally and vertically among adjacent cells, such that the h_{max} heuristic estimates the same values as the *Manhattan* heuristic.
- (ii). Describe briefly in STRIPS how to model the problem where a robot can move horizontally and vertically among adjacent cells, such that the h_{max} heuristic estimates the same values as the *Optimal* heuristic.

- (iii). Using your last STRIPS encoding where $h_{max} = h^*$, an initial state $s_0 = \text{robot at location } I$ and a goal state $s_g = \text{robot at location } K$, compute $h_{max}(s_g)$ and h_{ff} from the best supporters induced by h_{max} .
- (b). We have 2 rooms A and B , 2 objects o_1 and o_2 . 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 o_1 and o_2 into room B , given that both are initially at A .
- (i). Model the problem in STRIPS in such a way that the optimal plan would be the following:
- $$\text{pick}(o_1, A), \text{move}(A, B), \text{drop}(o_1, B), \text{move}(B, A), \text{pick}(o_2, A), \text{move}(A, B), \text{drop}(o_2, B)$$
- (ii). What's the $h_{max}(s_0)$ value of your STRIPS model, where s_0 stands for the initial state?

Question 3

(10 Marks)

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 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 received a reward of 5, or goes back to the *UnknownCereal* state with probability 0.7 and receives a reward of -2.

Answer the following questions about this MDP:

- (a). [5 marks] Assuming a discount factor $\gamma = 0.9$, 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 | 2 | 3 |
|------------------|---|-----|---|---|
| V(KnownCereal) | = | 0.0 | | |
| V(UnknownCereal) | = | 0.0 | | |

- (b). [5 marks] Given the value function that you calculate above, what policy would maximise the robot's expected reward? Show your working.

Question 4

(6 marks)

Explain the concept of an *all-outcomes determinisation* in non-deterministic planning, and how it is used to solve non-deterministic planning problems.

Question 5

(6 Marks)

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.

Question 6

(12 Marks)

- (a). [6 marks] 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.

- (b). [6 marks]

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 A chooses IN and 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.

Question 7

(6 Marks)

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.

Question 8

(6 Marks)

This is an exercise to help develop your intuition about why TD methods are often more efficient than Monte Carlo methods. Consider the following driving home scenario:

“ Suppose you have lots of experience driving home from work. Then you move to a new building and a new parking lot (but you still enter the highway at the same place). Now you are starting to learn predictions for the new building. ”

- (a). Would TD update be better on average than an Monte Carlo update on this scenario?
- (b). Give another example scenario — a description of past experience and a current state — where the best update (TD or MC) would be the opposite of the driving scenario. Can you see why TD updates are likely to be much better, at least initially, in some scenarios? Justify your answers.

Question 9

(5 marks)

Below is a tree from MCTS in which 100 roll-outs have been performed, and 7 nodes expanded. The notation X/Y indicates that Y number of roll-outs have been performed, with a cumulative score of X , and thus X/Y is the average score and value for that state.

Given this tree, what would action should be chosen at state S to maximise expected reward?

