AY2122 Sem2
CS4246 and CS5446: AI Planning and Decision Making
Final Assessment

# of Questions: 12
Duration: 90 minutes
Total Exam Points: 30.00

NATIONAL UNIVERSITY OF SINGAPORE

CS4246/CS5446 - AI Planning and Decision Making

Semester 2 : AY 2021/22

Time Allowed: 90 minutes

INSTRUCTIONS TO STUDENTS

1. This assessment contains **12** questions.

2. Please answer **ALL** questions.

3. This is a **NON-SECURE BLOCK INTERNET** assessment.

4. Permitted materials: Legal softcopy of the module textbook, homework/tutorial solutions, lecture notes, self-created notes. Hardcopy of the module textbook, homework/tutorial solutions, lecture notes, handwritten/self-created notes.

5. Permitted devices: NUS approved electronic calculators

6. You are not permitted to leave the venue in the first hour of the assessment as well as in the last 15 minutes.

7. Please place your student card or identification document (NRIC, driving license, etc.) on the top right hand corner of your desk.

8. Please switch off your personal devices with communication features and leave it on the floor next to your desk at all times.

9. If you wish to communicate with an invigilator, go to the washroom, or leave before the end of the assessment, please raise your hand to inform the invigilator.

10. You will be liable for disciplinary action which may result in expulsion from the University if you are found to have contravened any of the clauses below,

- Violation of the NUS Code of Student Conduct (in particular the part on Academic, Professional and Personal Integrity), NUS IT Acceptable Use Policy or NUS Examination rules.

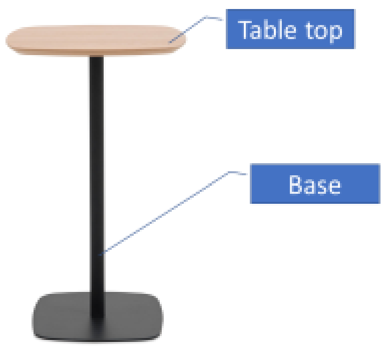
- Possession of unauthorized materials/electronic devices.
- Bringing your mobile phone or any storage/communication device with you to the washroom.
- Unauthorized communication e.g. with another student.
- Reproduction of any exam materials outside of the exam venue.
- Photography or videography within the exam venue.
- Plagiarism, giving or receiving unauthorised assistance in academic work, or other forms of academic dishonesty.

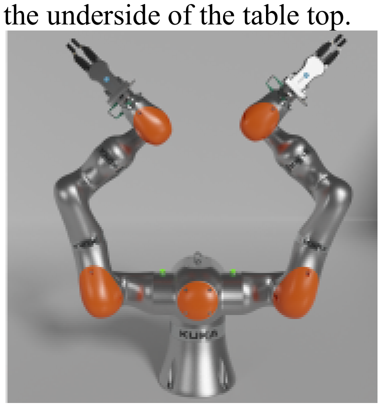
11. Once you have <u>completed</u> the assessment,

- Click on the "Exam Controls" button and choose "Submit Exam".
- Check off “I am ready to exit my exam” and click on "Exit" to upload your answers to the server.
- You will see a green confirmation window on your screen when the upload is successful. **Please keep this window on your screen.**
- If you do not see a green window, please disconnect and reconnect your WIFI and try again.
- Please be reminded that it is your responsibility to ensure that you have uploaded your answers to the Software.
- Please remain seated quietly in the meantime.

Question #: 1

[Total: 8 marks] Note: This question has multiple parts. Scroll down to read the full question. Please write the part number correctly in your answer.
Consider a 2-arm robot assembling a table having two parts as show in the diagram. The robot can *flip* items (e.g., to turn the item upside down and place on the floor). The robot can *attach* one part to the other. The robot needs to *get* the item first to assemble it. The assembly always happens such that:
i. one item is on the ground and the other item is attached to the item on the ground. (E.g., if we need to assemble any two items A and B, A is appropriately positioned on the ground and B is attached to it)
ii. the larger item is placed on the ground and the smaller item is attached to it (for reference the base is smaller than the table top).
iii. the base can be attached only on the underside of the table top.





Use the following predicates in your answers (where necessary)

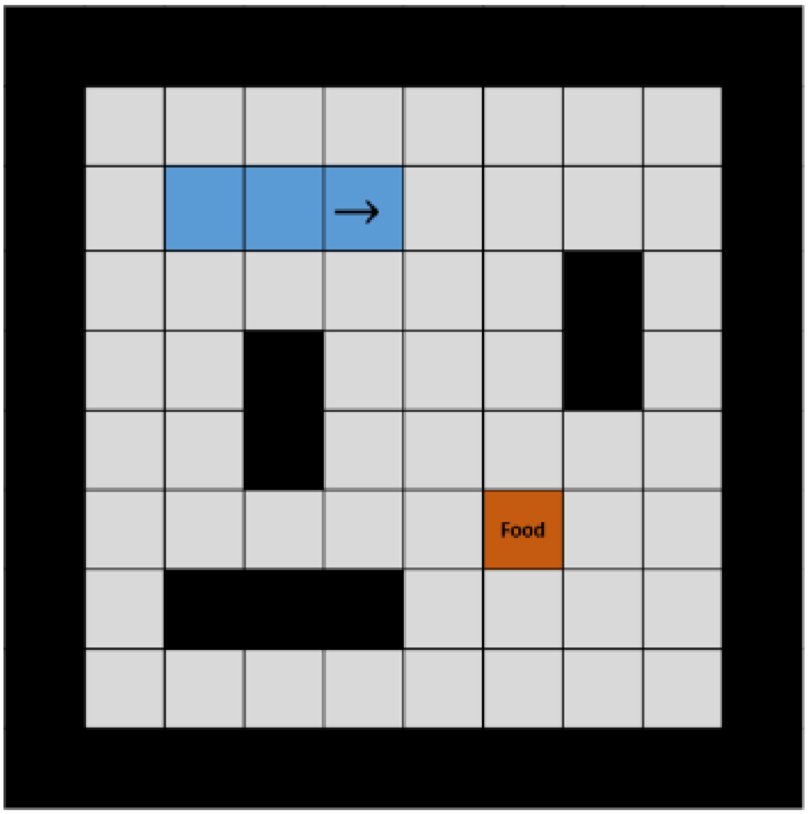
- UpsideDown(x) –entity x is upside down; facing down
- FacingUp(x) –entity x is facing up
- Table(table) –table is an assembled table

Assume all the table parts are on the floor (facing up where it matters) to begin with. Treat the robot as one entity. The robot is on the floor and can reach all the parts easily. Robot can hold only one item at once and needs both hands to hold. Once placed on the floor, the item is considered stable and there is no need to hold it anymore. Assume STRIPS representation. However, the following extensions are available: inequality operation is available and action Attach will make the predicate Table(table) to be true.
Questions:
a) [3 marks] Write out the initial state of the problem. Clearly define all additional predicates and fluents as necessary that you will want to use in the solution.
b) [3 marks] Write the action schema for the flip, attach, and get actions. (Hint: You may need more than one variant of the flip action.)
c) [1 mark] List the action sequence that would solve the problem of assembling the table starting from the initial state.
d) [1 mark] What is the goal state of the problem?
Notation: Use full words (case doesn't matter) to represent constants (e.g., Robot or robot) and lower-case letters (e.g., r) to represent variables in your answer. Use &, ||, ~ for AND, OR, and NOT respectively.
If you make additional assumptions, state them clearly in the answer.

Item Weight: 8.0

Question #: 2

[Total: 6 marks] Note: This question has multiple parts. Scroll down to read the full question. Please write the part number correctly in your answer.
Snake is a popular game of yester-years that was conceptualized in 1976 arcade game Blockade. The player controls the "snake" (shown in blue in the image below, the arrow indicates the head of the snake) by moving left, right, up, and down (which are the available actions). The snake continues to move in the direction pointed to by the head until a new direction is set using the actions mentioned above. The snake grows in length by one cell when it eats food (shown in orange). New food spawns as soon as the current food on the screen is consumed. The location where food spawns is stochastic. The snake's own body, and the walls in the environment (shown as black cells) are the obstacles. The interior walls can be present anywhere in the environment, shown here is just an example. The game ends when the snake head touches any of the obstacles. The size of the environment can range from 8x8 to 100x100 cells.



You are asked to model the snake game as an MDP in the reinforcement learning setting.
Answer the following questions based on the description above.
a) [4 marks] Identify the features (or state variables) that can represent the state of the game accurately and succinctly.
Note: You are not allowed to use visual features, i.e., image/video or grid bit-map to represent the state.
Write the state variables and a one-line description of what each variable is used to model in the game in your answer as a bullet list.
b) [2 marks] Define a simple reward function that can be used in this problem.

Item Weight: 6.0

Question #: 3

[Total: 6 marks] Note: This question has multiple parts. Scroll down to read the full question. Please write the part number correctly in your answer.
Consider the payoff matrix for a 2-player game. For simplicity, we will consider the actions to be A and B. The first value indicates payoff for the row player and the second for the column player.

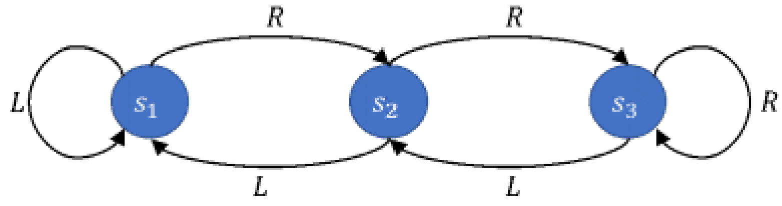
	A	B
A	5, 8	7, 12
B	7, 10	3, 4

a) [2 marks] Does there exist pure strategy Nash equilibrium for this game? (State yes or no explicitly in addition to the explanation)
If so, what is/are the strategy profile(s) for the pure strategy Nash equilibrium?
If not, explain why is there no pure strategy Nash equilibrium.
b) [2 marks] What are the mixed strategies of the row player and the column player? Write in [probability: action, probability: action] form.
c) [2 marks] What is the expected utility of the row player if both players are following their mixed strategies? Write your answer correct to 2 decimal places.

Item Weight: 6.0

Question #: 4

[2 marks] Consider a 3-state POMDP as shown below:



The states are s1, s2, and s3. There are two actions available R and L (representing right and left respectively). The actions have 90% success rate, and when the action fails, the agents moves in the opposite direction. When there is no more state to go Left (or Right), the agent stays put in the current state (e.g., executing L in s1 will result in the agent remaining in s1).
s3 is the goal state.
The agent receives observation indicating if the agent is in the goal state. However, the observation is correct only 70% of the time.
The initial belief is as shown: b0 = [1/2, 1/2, 0], i.e., uniform probability of being in either of the non-terminal states.
Calculate the belief b1 after the agent executes action R and receives the observation.
Write your answer in the form b1= [b1(s1), b1(s2), b1(s3)] correct to five decimal places. You don't have to show the computation in the answer.

Item Weight: 2.0

Question #: 5 to 12 MCQ (not archived)