CS 188 Spring 2021 | Introduction to Artificial Intelligence | Written HW 2

Due: Wednesday 2/24/2021 at 10:59pm (submit via Gradescope).

Policy: Can be solved in groups (acknowledge collaborators) but must be written up individually

Submission: (UPDATED INSTRUCTIONS) It is recommended that your submission be a PDF that matches this template. You may also fill out this template digitally (e.g. using a tablet. **However, if you do not use this template, you will still need to write down the below four fields on the first page of your submission.**)

First name	
Last name	
SID	
Collaborators	

For staff use only:

Q1.	Expectimax Yahtzee	/40
Q2.	Propositional Logic	/40
Q3.	Programming and Visualizing AI Game Agents	/20
	Total	/100

Q1. [40 pts] Expectimax Yahtzee

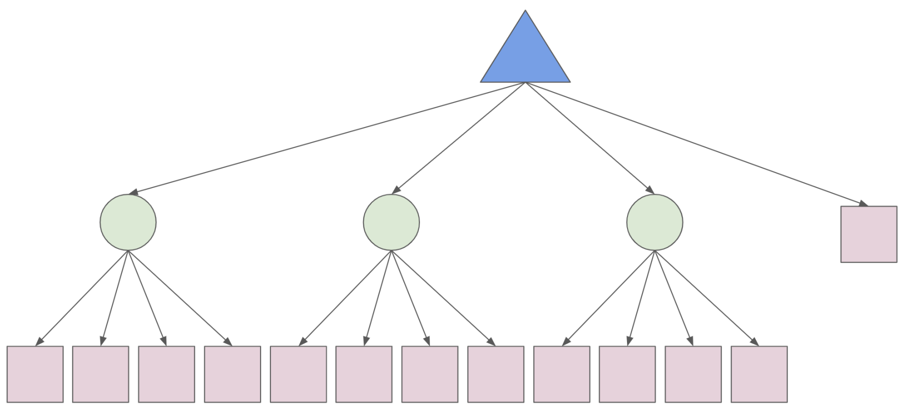
Consider a simplified version of the game *Yahtzee*. In this game, we have 3 regular tetrahedral dice with 4 sides each (numbered 1-4) and the game begins by rolling all 3 dice. At this point, a player can make a decision: pick one of the 3 dice to reroll, or don't reroll anything. Then, points are assigned as follows:

- A reward of 10 points is given for two-of-a-kind.
- A reward of 15 is given to three-of-a-kind.
- A reward of 7 points is given for rolling a series (1-2-3 or 2-3-4).
- Otherwise (or if the sum is higher than the special reward), the score is equal to the sum of all 3 dice.

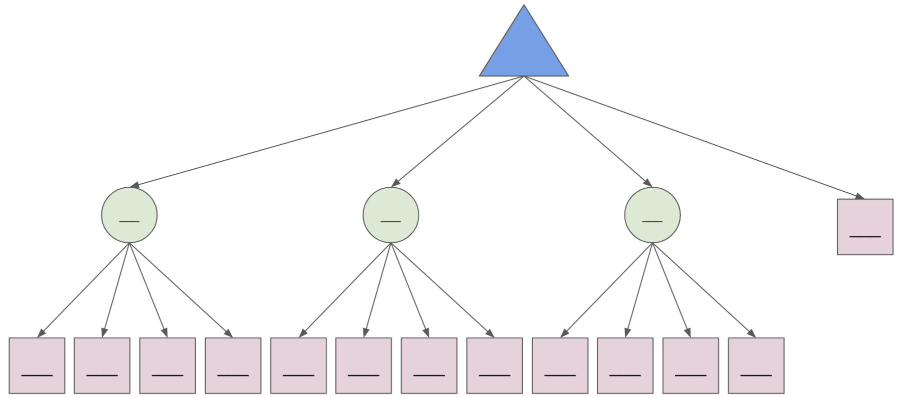
(a) We will formulate this problem as an expectimax tree.

(i) [3 pts] The resulting tree for the problem is drawn below. Given a specific initial roll, the branching factor (of the player's decision) from the root node is $\boxed{}$. The branching factor at the chance nodes is $\boxed{}$. What do those chance nodes represent? (There are multiple solutions, you only need to write down one solution)

- Chance node 1:
- Chance node 2:
- Chance node 3:



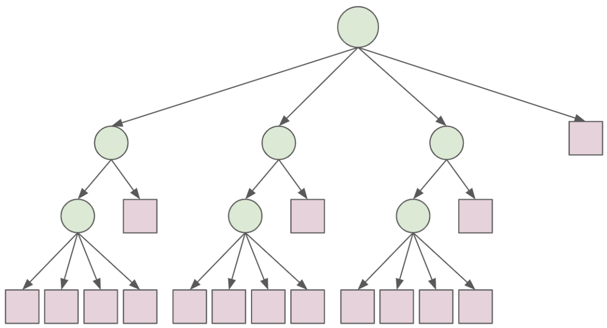
(ii) [7 pts] Given a starting roll (1,2,4) (corresponding to the outcomes of die rolls 1, 2, and 3 respectively), what move should you take? Fill in the values of the expectimax tree below to justify your answer.



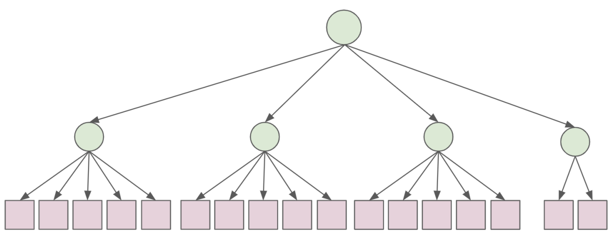
Now suppose the human player does not understand how to play the game, and as a result, they choose any action with uniform probability, regardless of the initial roll. Moreover, we assume that the human's choice will be carried out by a "somewhat helpful" robot called Albertbot: given a configuration of dice and the desired action from the human, this robot either actually implements the human's action (with probability $1-p$) or overrides it with a 'no reroll' action (with probability $p > 0$). If the human action is already 'no reroll', then the robot does not interfere.

(b) Given a particular Yahtzee roll *roll*, Let A, B, C and D be the expected reward of performing actions 'reroll die 1', 'reroll die 2', 'reroll die 3', and 'no reroll', respectively.

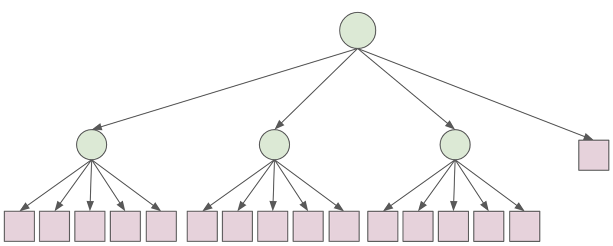
(i) [3 pts] Which of the following trees best represents the expectimax tree, after accounting for the presence of the Albertbot?



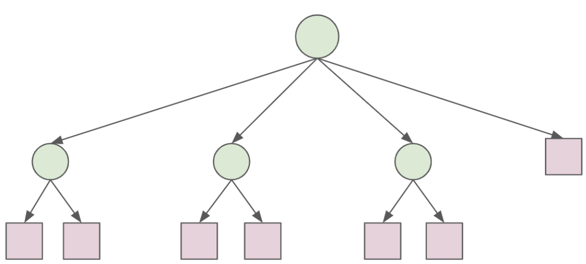
○



○



○



○

○ None of the above.

(ii) [5 pts] Express R_H and R_{AH} in terms of A, B, C and D, where:

- R_H is the expected reward for the human acting *without* Albertbot's help.
- R_{AH} is the expected reward for the human acting *with* Albertbot's help.

Show all steps of your work and **write your expression into the form of** $X + Yp$, where X and Y are expressions that contain A, B, C and D but not p.

$R_H =$

$R_{AH} =$

(iii) [5 pts] What is the condition for our Albertbot to strictly increase expected reward? Write the condition above using only $A, B, C, D, >$.

(iv) [2 pts] In one sentence, please describe the situation when the condition above is true.

(c) Your friend Diana argues that a helpful robot should not only override the human player's "reroll" choice with probability p (and replace it with a "no reroll"), but also override the human player's "no reroll" choice with probability p (and replace it with the outcome of selecting one of the 3 dice at random and rerolling that dice). Diana needs your help with drawing the new expectimax tree for the Dianabot.

(i) [5 pts] Draw the expectimax tree for Dianabot. You need to draw out the tree with all nodes in their correct shapes; you do not need to label any values in the tree. Hint: you can start by modifying the expectimax tree for the Albertbot.

(ii) [5 pts] What is the expected reward for a random human player with Dianabot's assistance? Again, please **show all steps of your work** and **write your expression into the form of** $X + Yp$, where X and Y are expressions that contain A, B, C and D but not p.

$R_{DH} =$

(iii) [5 pts] Under what condition on A, B, C, D is the human better off using Dianabot rather than Albertbot?

Q2. [40 pts] Propositional Logic

(a) Consider a propositional language with 4 symbols: A, B, C, D. For each of the following sentences, mark how many models satisfy the sentence out of the 16 possible models.

(i) [3 pts] $\alpha_1 = A \vee B$:

(ii) [3 pts] $\alpha_2 = (A \wedge B) \Rightarrow C$:

(iii) [4 pts] $\alpha_3 = (A \wedge B) \vee (\neg C \vee D)$:

(b) Suppose there are three chairs in a row, labeled L(eft), M(iddle), R(ight) and three persons A, B, and C. Everyone has to sit down but, unfortunately,

- A doesn't want to sit next to B
- A doesn't want to sit in the left chair
- C doesn't want to sit to the right of B

We will formulate these constraints in propositional logic using only variable $X_{p,c}$ to mean that person p sits in chair c. **Please express the constraints in CNF.**

(i) [2 pts] A doesn't want to sit next to B.

(ii) [1 pt] A doesn't want to sit in the left chair.

(iii) [1 pt] C doesn't want to sit to the right of B.

(iv) [4 pts] Are there other constraints? If yes, express them in propositional logic.

(v) [2 pts] Lastly, can we satisfy these constraints? (A yes/no answer with some justification is sufficient for the problem. You do not need a formal proof.)

(c) Logical Inference

Given

$$KB = (A, A \Rightarrow B, A \Rightarrow C, B \wedge C \Rightarrow D)$$

(i) [4 pts] Show the steps in a forward chaining algorithm for proving $KB \models D$.

(ii) [2 pts] How do we prove $KB \models D$ using a SAT solver?

(iii) [4 pts] Write out the necessary clauses in CNF representation of the sentence required for (b).

(iv) [10 pts] Show the steps in the operation of DPLL, assuming a fixed variable ordering (A, B, C, D) and a fixed value ordering (true before false). Remember to apply early termination, pure literals (repeatedly), and unit clauses (repeatedly), keeping track of which clauses have already been satisfied in the process.

Q3. [20 pts] Programming and Visualizing AI Game Agents

(a) [10 pts] Please make a copy of the colab linked on piazza. On your copy, complete the following functions. If your code passes, you will see "Validate: Pass" for parts a and b. (Hint: the staff solution has just two lines for each of the below coding subparts):

(i) [3 pts] `minimax()`. Here you will complete the implementation for `min_player` and `max_player`. You can get the value of the ith child by calling `tree.get_value(children[i])`. Remember to call `tree.set_value(node, value)` which sets `node`, a parameter of the function, to the value `value`.

(ii) [3 pts] `non_optimal_min_player()`. Here you will complete the implementation for `min_player_strategy_id = 3`, which is essentially a chance node. You will be calling `tree.get_value()` and `tree.set_value()`, just like in the previous part.

(iii) [4 pts] `tic_tac_toe_AI()`. First instantiate a `tictactoe_game_tree`. Think about what sort of game search (expectimax, minimax, non-zero sum) should you call, and call this pre-defined algorithm in the function. Note that after your lines of code are written, the variable `move` should be assigned to a tuple, which indicates the first move the maximizing agent should take.

(iv) [0 pts] **Submission Directions**: On your colab, please go to `File` → `Download .py`, and submit this python file to gradescope. We will set up an autograder for the coding portion after the due date. Note that the autograder will not run if you do not submit a python file!

(b) [1 pt] What is the root value of the tree (a maximizer node) when the min agent acts optimally? (Write the value of the root of the visualized game tree when `min_player_strategy_id = -1`.)

(c) [4 pts] These are the different strategies of the "min_player" visualized in the colab notebook. Please write a sentence describing the distribution of the root value (by looking at the histogram). In your description, estimate the mean and talk about the modality (unimodal, bimodal, etc.):

(i) [1 pt] `min_player_strategy_id = 0`, where "min" chooses the first child regardless of its value

(ii) [1 pt] `min_player_strategy_id = 1`, where "min" chooses the 2nd smallest value among it's children

(iii) [1 pt] `min_player_strategy_id = 2`, where "min" chooses the largest child

(iv) [1 pt] `min_player_strategy_id = 3`, where "min" acts like a chance node (averages its children)

(d) [3 pts] Compare the root node mean of the four suboptimal "min" agents for part C to the value you wrote for part B (where "min" acts optimally). Give a proof for why a suboptimal "min" agent always increases the value of the root maximizer node, compared to a minimax tree.

(e) [1 pt] Give an intuitive explanation for why the expectimax agent (`min_player_strategy_id = 3`) achieves a lower root value on average than strategies 0 and 2.

(f) [1 pt] At the end of the Colab, you saw a rollout of the tic-tac-toe game where both players act optimally. Why does optimal player 2 ("X") place the piece in the center after player 1 ("O")'s first move? Give an intuitive explanation. It may help to consider what would happen if X, going after O's first move, did not move the piece on the center.