TDT4136 Exam 2023 Solution

Xavier F.C. Sánchez Díaz

January 2, 2024

Overview

This documents contains the solutions of the final (published) version of the final exam for Autumn 2023.

- There might be multiple correct answers in many of the questions (especially in the *short questions* section). These are only some of the available approaches.
- For *short questions*, the original question comes in *italics* and the answer is in normal font style.
  - The answers for all questions (except the last one) were provided by Pinar, as in the original word document.
  - The answer about the final question about individual fairness in AI was provided by Gleb, as in the original HTML document.
- For the rest of the tasks, the start of solutions are marked as **Solution**, and the end is marked with a square like this □

---

1 Short questions

Assume a Hill Climbing algorithm found a state from which none of the neighbour states produce an increase in the score value. What can this state be, if not a global maximum? What method/trick may be added to the algorithm in order to proceed from such states?

Answer: it may also be a local minimum or a plateau. Random start would be a good trick.

An AI system is used at the Human Resources (HR) department in the recruitment process. The system is given a job advertisement together with the applications of the candidates and it makes a suggestion about who to hire. Assume that the system is taking "corrective decisions", which means that it takes into consideration the social injustice issues that appear because of the bias in the past decisions and attempts to decrease injustice. Is this an episodic or a sequential environment? Justify your answer.

Answer: fully observable, and sequential.

This question is about the environment and agent matching. To behave intelligently in a partially observable environment, in which respects the agent should be different from a simple reflex agent?

Answer: needs memory and its action transition function operates with belief states , i.e., not single state.

If $h_1(s)$ and $h_2(s)$ are two admissible A heuristics, then would their average $h_3(s) = 1/2h_1(s) + 1/2h_2(s)$ be admissible? Explain your answer very briefly.*

Answer: Yes, as this is just a scalar multiplication/linear transformation.

A robot is going to help to move a person from one apartment to another one, by loading the ready boxes in different room to a truck parked outside of the building. There are 4 rooms in total, called R1,. . . .R4 where, in the beginning R1 has 3 boxes, R2 has 1 box, R3 has 5, and R4 has 2 boxes. On each turn, the robot can move a box into an adjacent room, in any direction: north, south, east, west. Some of the rooms are adjacent to another(s) while some others are not connected. In the beginning, the robot is in the truck. This can be represented as a search problem. How would you formulate a state in the search space in general, and how would the beginning state be represented?

Answer: A tuple $(agent, R_1, R_2, R_3, R_4, Truck)$, where $agent$ can have any value $a \in \{R_1, R_2, R_3, R_4, Truck\}$ is the location of the agent, and the others are numbers ≥ 0 telling how many boxes are in that respective room. The start state can then be represented as $(Truck, 3, 1, 5, 2, 0)$.

Aristotle argued that actions are justified by a logical connection between goal and knowledge of the action's outcome. Over 2000 years later his ideas were implemented in AI by two pioneer AI researchers in AI. Who are/were they and what is the name of the program they designed and coded?

Answer: Herbert and Simon. General Problem Solver

This question was in Assignment 1 and way fewer students than I wished gave good answers to this question. Now I hope everybody knows these 2 pioneers in the AI field.

I decided to give gave 1 point to everybody (i.e. like decreasing the weight of this question) and 2 points to those who gave correct answer.

What is the difference between stochastic and non-deterministic environments — according to the textbook?

Answer: Stochastic is when the probability of possible outcomes are explicitly dealt with, while in non-deterministic the possibilities are known but not quantified.

In some competitive multiagent environments, randomized behaviour is rational. Why, explain.

Answer: randomization makes predictability difficult.

Backward and forward chaining algorithms in logic works with sentences in a specific form, and a specific inference rule. What are the names of the representation and the rule?

Answer: Horn clauses and modus ponens.

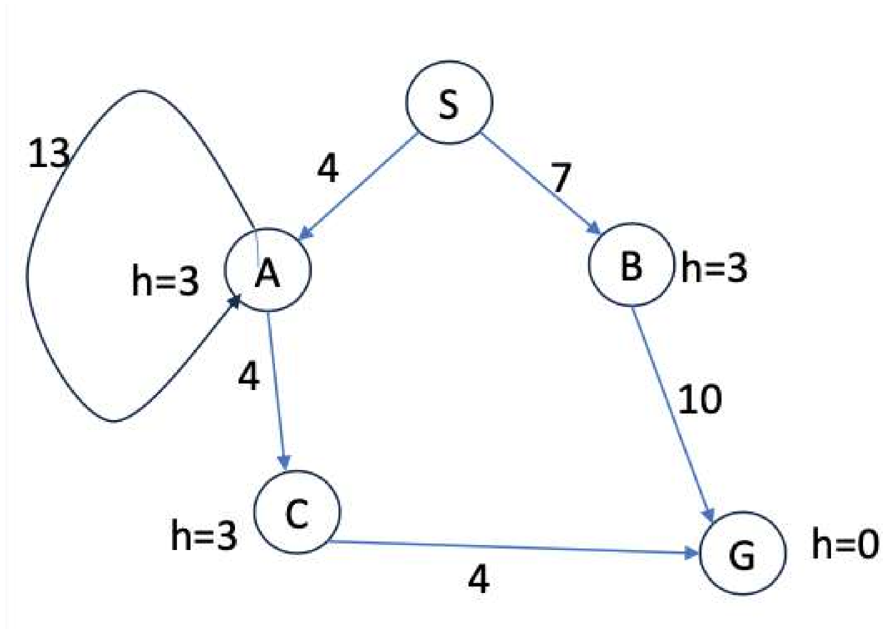

Figure 1: Graph for question 2.1: search algorithms

Describe what individual fairness means with respect to ethics in AI, and explain what kind of methods can be used to implement/realize it in an AI system.

Answer: Candidates are treated similarly to other candidates, regardless of what class they are. This requires a way to find candidates that are similar and checking whether hiring decisions are the same. If not, the system is either biased or inconsistent. A possible method for this is a transparent/traceable/explainable method like Case Based Reasoning or a Nearest Neighbour classifier.

2 Search (20 pts total)

2.1 Search algorithms (10 pts)

1. Execute tree-like search through the following graph (see Fig. 1) for the given algorithms below. Tie breaking: Left-to-right order of successors.

a- Breadth first search, b- Depth First search, c- Uniform cost search, d- Greedy best-first search, e- A* search

For each of the search algorithm,

- show the order in which the nodes are expanded.
- show the path from start to goal, or write "None" if no path to goal is found, and
- write the cost of the path found, in the following format

2.1.1 Solution - Search algorithms

First of all, note that "tree-like" search means no checking of loops- according to our textbook. However, the answers that considered both checking of expanded nodes and not checking them and writing about loop and not finding a path for Depth First and Greedy best methods are accepted as correct answer.

Unexpectedly many students did wrong for the breadth-first search - and this was not only about the early goal check.

BFS

- Order of expansion: S, A, B, (G)
- Found path: SBG
- Path cost: $10 + 7 = 17$

DFS

- Order of expansion: S, A, A, A, ...
- Found path: $\emptyset$
- Path cost: $\emptyset$

Uniform cost search (best cost first)

- Order of expansion: S, A, B, C, (G)
- Found path: SACG
- Path cost: $4 + 4 + 4 = 12$

Greedy best-first (best guess first)

- Order of expansion: S, A, A, A, ...
- Found path: $\emptyset$
- Path cost: $\emptyset$

A-star

- Order of expansion: S, A, B, C, (G)
- Path: SACG
- Path cost: $4 + 4 + 4 = 12$

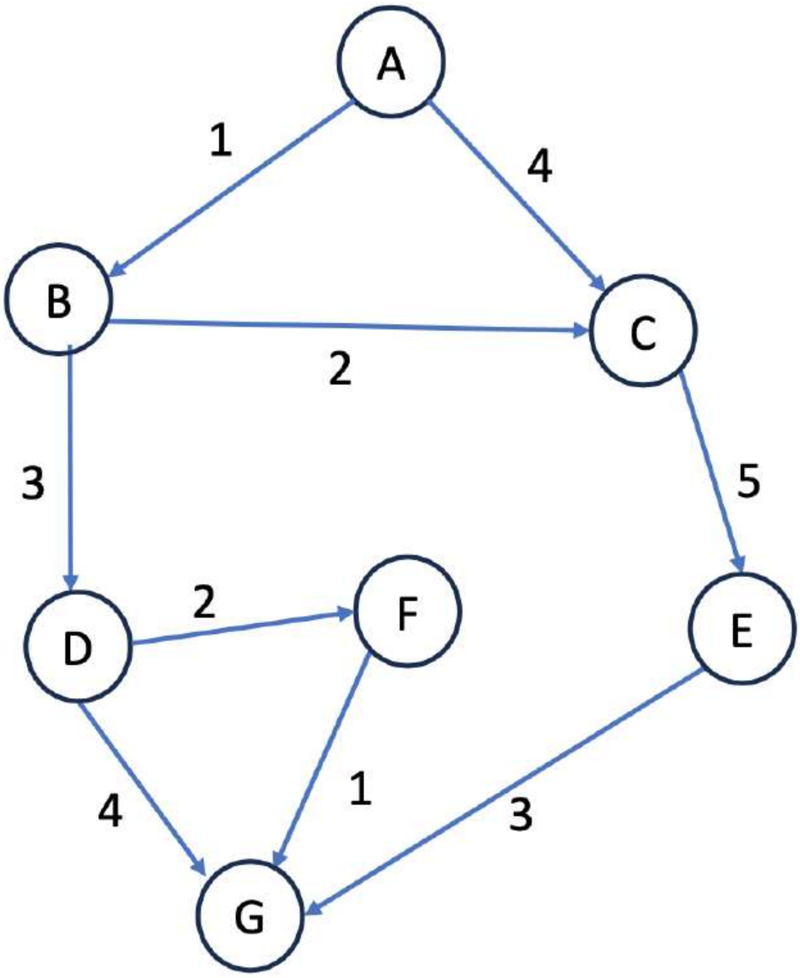

Figure 2: Graph for question 2.2: Consistency

2.2 A* consistency (10 pts)

Assume:

- $h(A) = 5$
- $h(C) = 4$
- $h(D) = 3$
- $h(E) = 3$
- $h(F) = 1$
- $h(G) = 0$

1. What values of $h(B)$ make h monotone/consistent?
2. What values of $h(B)$ will cause A* search to expand node A, C, B in this order?
3. What values of $h(B)$ will cause A* to return a sub-optimal path?

2.2.1 Solution - A* consistency

1) What values of $h(B)$ make h consistent?

- $h(B) \leq c(D) + h(D)$ so $h(B) \leq 3 + 3$ so $h(B) \leq 6$
- $h(B) \leq c(C) + h(C)$ so $h(B) \leq (2 + 1) + 4$ so $h(B) \leq 7$
- $h(A) \leq c(B) + h(B)$ so $5 \leq 1 + h(B)$ so $4 \leq h(B)$

Which means that $4 \leq h(B) \leq 6$ $\square$

2) What values of $h(B)$ makes A-star to expand "ACB"?

In order to expand C before, then $c(B) + h(B) > c(C) + h(C)$:

- $1 + h(B) > 4 + 4$ so $h(B) > 8 - 1$ so $h(B) > 7$ With $h(B) > 7$ A* would prefer expanding C first.

When on C, then it also needs to prefer B over E, so that means:

- $c(B) + h(B) \leq c(E) + h(E)$ so $1 + h(B) \leq (4 + 5) + 3$ so $h(B) < 11$

which means that $7 < h(B) < 11$ $\square$

3) What values of $h(B)$ will cause A-star to return a sub-optimal path?

Optimal cost is 7, through $ABDFG$. The only way for the path to be sub-optimal (given that we only change values of B) is to prefer C over B, and then E over B again. So $1 + h(B)$ needs to be strictly higher (assuming alphanumeric tie breaking) than both 8 and 11.

So $h(B) > 11$ $\square$

3 Logic (20 pts total)

3.1 Propositional logic (10 pts)

You are given the task of finding a book hidden somewhere in the house or the garden. You are given the following set of hints:

1. If the house is painted red, then the book is not in the kitchen.
2. If there is an apple tree in the front yard, then the book is in the kitchen
3. The house is painted red.
4. The tree in the front yard is an apple tree or the book is under the flower pot.
5. If the tree in the backyard is a plum tree, then the book is in the garage.

Question: Where is the book?

- a) Translate these hints into propositional logic sentences and show each proposition with a capital letter. e.g.,

```
A: The house is painted red.
.
.
F:...
```

- b) Using the inference rules provided below find out where the book is hidden. You may use only the inference rules in the given list but it is not necessary to use all the rules in the list, and you can use a rule more than once if needed. **Constraint: Your solution must use Modus tollens**.

In your answer refer to the name of the rule you are using in each step. Give a number to each step, and write in each step: which of the propositional sentences (A, B...) are used, the name of the inference rule used, and the derived sentence (also give a new capital letter to the derived sentence which may in turn be used in one of the next steps).

3.1.1 Solution - Propositional Logic

1. If the house is painted red, then the book is not in the kitchen
2. If there is an apple tree in the front yard, then the book is in the kitchen
3. the house is painted red
4. the tree in the front yard is an apple tree or the book is under the flower pot
5. If the tree in the back yard is a plum tree, then the book is in the garage

- R the house is red

- K the book is in the kitchen
- A there is an apple tree in the front yard
- F the book is under the flower pot
- P the tree in the back yard is a plum tree
- G the book is in the garage

Translation

1. $R \implies \neg K$
2. $A \implies K$
3. R
4. $A \vee F$
5. $P \implies G$

Inference
Modus Ponens: 1 and 3 yield 6

$$\frac{R \implies \neg K, R}{\neg K}$$

Modus Tollens: 2 and 6 yield 7

$$\frac{A \implies K, \neg K}{\neg A}$$

Disjunctive Syllogism: 4 and 7 yield 8

$$\frac{A \vee F, \neg A}{F}$$

Hence, the book is under the flower pot □

3.2 First Order logic (10 pts)

You are given the following information:

Fido is father of Snowy. Fido is alive. One's father is ones parent and vice versa. Somebody's alive parent is older than them.

Question: Is there anybody older than Snowy?

- a) Represent this information and the question in form of First Order Logic (FoL) sentences.
- b) Using resolution refutation answer if there is anybody older than Snowy? If there is, who is this?

Number the CNF sentences starting from 1, and then employ resolution refutation. Show how you convert the FOL sentences to CNF sentences.

3.2.1 Solution - First Order Logic

1. $father(fido, snowy)$
2. $alive(fido)$
3. $parent(X, Y) \leftrightarrow father(X, Y)$
4. $[parent(X, Y) \land alive(X)] \implies older(X, Y)$

Sentences 3 and 4 apply to all X and Y, so we can safely omit the quantifiers for inference.

CNF

1. $father(fido, snowy)$
2. $alive(fido)$
3. $\neg parent(X, Y) \lor father(X, Y)$
    1. $\neg father(X, Y) \lor parent(X, Y)$
4. $\neg parent(X, Y) \lor \neg alive(X) \lor older(X, Y)$

- Q: $\exists X[older(X, snowy)]$
- $\neg Q = \neg\exists X[older(X, snowy)]$. So:

$$\begin{aligned}\neg\exists X[older(X, snowy)] =& \\ &= \forall X[\neg older(X, snowy)] \\ &= \neg older(X, snowy)\end{aligned}$$

We can safely omit the universal quantifier.

Resolution

- Using $\neg Q$ and 4, substituting $Y \leftarrow snowy$, it yields 5
$$\frac{older(X, snowy), \neg parent(X, snowy) \lor \neg alive(X) \lor older(X, snowy)}{\neg parent(X, snowy) \lor \neg alive(X)}$$
- Using 5 and 2, substituting $X \leftarrow fido$, it yields 6
$$\frac{\neg parent(fido, snowy) \lor \neg alive(fido), alive(fido)}{\neg parent(fido, snowy)}$$
- Using 6 and 3.1, substituting $X \leftarrow fido, Y \leftarrow snowy$, it yields 7
$$\frac{\neg parent(fido, snowy), \neg father(fido, snowy) \lor parent(fido, snowy)}{\neg father(fido, snowy)}$$
- Using 7 and 1, it yields bottom:
$$\frac{\neg father(fido, snowy), father(fido, snowy)}{\bot}$$

Therefore our assumption about not existing someone older than Snowy is false, which means there is someone older than Snowy. A possible value is Fido.

4 CSP (15 pts total)

4.1 CSP

Gardeners Adrian, Brian and Celine have 2 hours to complete 5 tasks (1,2,3,4,5) in NTNU's gardens. Each task takes 1 hour to finish and each of them can work on only one task at a time, and alone. Each gardener has different expertise areas and can perform only a subset of the tasks, as shown below:

- Adrian: Tasks 1,2,3
- Brian: 1,2,5
- Celine: 2,4,5

They have the following restrictions:

- Task 1 must be completed before Task 2
- Task 3 must be completed before Task 5

This problem can be formulated as a constraint satisfaction problem using one variable for each task: X1, X2, . . ., X5 of which possible values are a subset of {A1, A2, B1, B2, C1, C2}, where X1= B2 means that Task 1 is performed by Brian using time slot 2. The domain of each variable is shown below:

- X1: A1, A2, B1, B2
- X2: A1, A2, B1, B2, C1, C2 – i.e, any of the gardeners can do this task
- X3: A1, A2
- X4: C1, C2 – i.e., only Celine can do this task
- X5: B1, B2, C1, C2

1. Write the constraints (2 points).
2. Is the initial state arc consistent? If not, use AC3 in order to prune the values of the variables and write down the pruned domain values for each variable (6 points)
3. Solve this problem (the reduced one) using backtracking – without forward checking. As heuristics use the minimum remaining values for variable ordering (tie breaking according to numerical order) and least constraining value for value ordering (tie breaking according to alpha-numerical ordering). Write down the solution, i.e., variables and their value, in the order the variables assigned a value.

4.1.1 Solution - CSP

$X = \{X_i : i \in 1..5\}$ which are the tasks - Domain is $D_i \in D = G \times T$ where - $G = \{A, B, C\}$ which are the gardeners - $T = \{1, 2\}$ which are the time slots - Each D_i is restricted. See below.

- $D_1 = \{A1, A2, B1, B2\}$
- $D_2 = \{A1, A2, B1, B2, C1, C2\}$
- $D_3 = \{A1, A2\}$
- $D_4 = \{C1, C2\}$
- $D_5 = \{B1, B2, C1, C2\}$
- $C = \{X_1 \ll X_2, X_3 \ll X_5, all\_diff\}$

Is the initial state arc consistent?

No it isn't, as there are conflicts. Let's start with min values, D_3:

D_3 needs to be BEFORE D_5, so no value in D_5 allows $D_3 = A2$. We remove it and have $D_3 = \{A1\}$. The opposite holds, so we remove $B1$ and $C1$ from D_5, resulting in $D_5 = \{B2, C2\}$

The same applies to D_1: no value in D_2 complies with the fact that $X_1 \ll X_2$, so we remove $A2$ and $B2$ from D_1, to obtain $D_1 = \{A1, B1\}$, and $A1, B1$ and $C1$ from the domain of D_2: $D_2 = \{A2, B2, C2\}$.

The last pruning is done between D_1 and D_3, considering the all_diff constraint. We remove $A1$ from $D_1/$. At the end we have:

- $D_1 = \{B1\}$
- $D_2 = \{A2, B2, C2\}$
- $D_3 = \{A1\}$
- $D_4 = \{C1, C2\}$
- $D_5 = \{B2, C2\}$

Solve the new reduced problem

Using **backtracking** with **no inference** using `min_values` (MRV) and `min_conflicts`:

1. $X_1 \leftarrow B1$
2. $X_3 \leftarrow A1$ (`min_conflicts`)
3. $X_4 \leftarrow C1$ (`min_conflicts`)
4. $X_5 \leftarrow B2$ (`min_conflicts`)
5. $X_2 \leftarrow A2$ (`min_conflicts`)

All of the values are different, X_1 is before X_2, and X_3 is before X_5 $\square$

5 Game theory (10 pts total)

5.1 Game Theory

Recent developments towards Artificial General Intelligence (AGI) and its potential benefits and risks made it an important issue on a national level. Countries are trying to decide whether to accelerate, maintain or hinder the development of AGI. Consider two countries, US and China, that need to decide whether to accelerate, maintain or hinder the development of AGI.

Table 1: The AI acceleration game

		China		
		Accelerate	Maintain	Hinder
	Accelerate	6 / 6	8 / 4	9 / 2
US	Maintain	4 / 8	7 / 7	10 / 2
	Hinder	2 / 9	2 / 10	5 / 5

1. Explain in 3-5 sentences why this decision requires strategical reasoning? (1%)
2. Does this game have a weakly or strongly dominant strategy? If it exists, explain how you found it. If not, explain why not (2%)
3. Apply iterated elimination of strictly dominated actions to find a game solution, showing each step of the process.
4. Does this game have a Nash equilibrium? If it exists, explain how you found it. Otherwise, explain why it does not exist (3%)
5. Which strategy maximizes social welfare? Explain how you found it.

5.1.1 Solution - Game Theory

Does this game have a weakly or strongly dominant strategy? No, there is no strategy that is better than the rest under all scenarios.

Game solution through iterated elimination of dominated actions

1. We eliminate first *Hinder* which is strongly dominated by the other strategies. This applies for both players, and both column 3 and row 3 are removed.

Table 2: First elimination of dominated strategies

		China	
		Accelerate	Maintain
	Accelerate	6 / 6	8 / 4
US	Maintain	4 / 8	7 / 7

2. We now eliminate *Maintain* which is strongly dominated by *Accelerate*. This also applies for both players, and both column 2 and row 2 are removed.

3. The solution is then ***Accelerate/Accelerate*** □

Nash Equilibrium

For each strategy in the column player, we get the best action for the row player and highlight. We then do the same, but now switching the players' positions.

Table 3: Nash equilibrium in the AI game

		China		
		Accelerate	Maintain	Hinder
	Accelerate	**6** / **6**	**8** / 4	9 / 2
US	Maintain	4 / **8**	7 / 7	**10** / 2
	Hinder	2 / 9	2 / **10**	5 / 5

The Nash equilibrium is ***Accelerate/Accelerate*** which contains the best strategy for both players.

Maximising social welfare

The sum of payoffs is highest at ***Maintain/Maintain***, which is the strategy maximising social welfare.

6 Adversarial search (5 pts total)

6.1 Adversarial search

Consider a scenario where a trade union and a company are negotiating about the yearly salary increase. The union decided to use one of the two strategies:

- Ask for what they are willing to accept - increase of 6%
- Ask for a much higher increase of 10% with the hope that the company will agree on something close to it.

From previous years, they know how the company behaves with low and high demands. When the demand is 6%, there is a 50% chance that the company will propose even lower pay increase: 3% or 4%. There is also 50% chance that the company will try to meet the demand, proposing 5% or 6% increase.

When demand is 10%, the company with 25% chance will consider it unrealistic and come back with lower proposals of 2% or 4%. There is also 75% chance that the company will consider it realistic and come back with proposal of 8% or 10%.

Your task is to use ExpectiMiniMax to come up with the best strategy for the trade union, either ask for 6% or ask for 10%.

1. Consider the tree above where the trade union is MAX (A) and the company is MIN (D-G). Fill in (write down) the values for all nodes (A-G, and d1-g2) as provided in the text (1%)

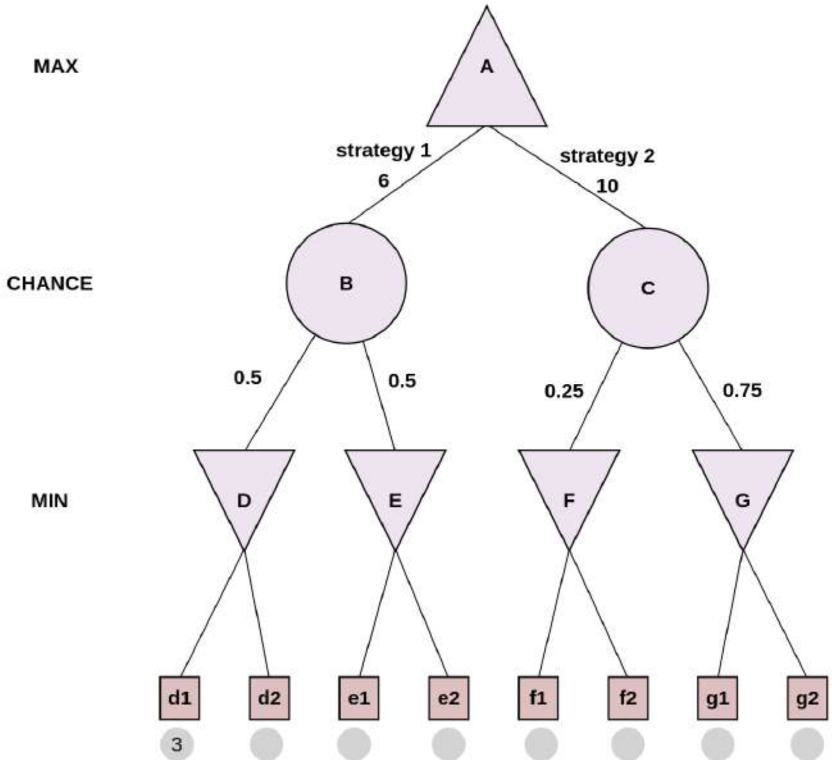

Figure 3: Adversarial search minmax tree

2. Compute ExpectMiniMax for all nodes except terminal nodes. Show your computations (3%)
3. Make a decision: which strategy is the best for the trade union? Explain how you reached that conclusion (1%)

6.1.1 Solution - Adversarial search

Node values

- d2 = 4
- e1 = 5
- e2 = 6
- f1 = 2
- f2 = 4
- g1 = 8
- g2 = 10

ExpectMiniMax values

- ExpectMiniMax(D) $= \min(3, 4) = 3$
- ExpectMiniMax(E) $= \min(5, 6) = 5$
- ExpectMiniMax(F) $= \min(2, 4) = 2$
- ExpectMiniMax(G) $= \min(8, 10) = 8$

And then chance nodes

- ExpectMiniMax(B) $= 0.5 * 3 + 0.5 * 5 = 4$
- ExpectMiniMax(C) $= 0.25 * 2 + 0.75 * 8 = 6.5$

And then game solution

- ExpectMiniMax(A) $= \max(4, 6.5) = 6.5$ □

Make a decision

Go with **strategy 2 (10% increase)** since it has higher expected return (6.5) than strategy 1 (6% increase) □

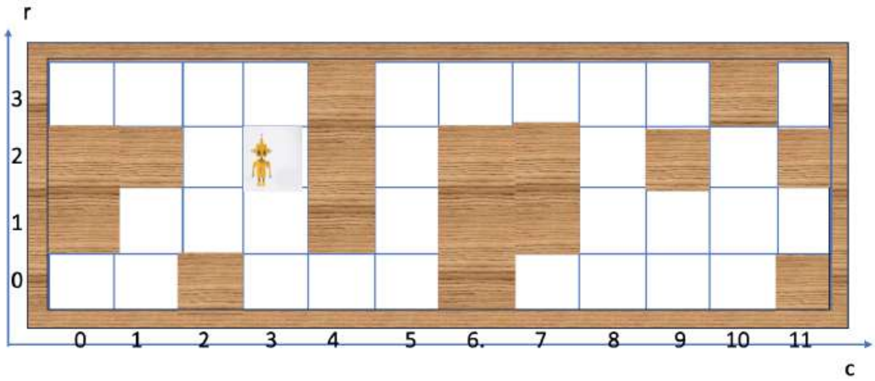

Figure 4: Robot in a partially observable environment

7 Partially observable environment (5 pts)

7.1 Partially observable environment

Assume a robot in a maze with walls, as shown in the Fig. 4. The robot can only see the walls but not its own locations (i.e-, which of the "squares" it is in the maze). A percept includes data about the existence of walls on the north, south, west and east of the square in which the robot currently is. For example. P1={North, South, NoWest, NoEast} means that the robot senses a wall in its north and south, and but not on east and west. The location of a square that the robot beliefs it may be in is shown with the row and column number of the square. For example, the robot in the figure is in location $S = (r2, c3)$. It has possible actions: Right, Left, Up, and Down.

1. If the robot's perception, at time step 1, is P1= { North, South, West, NoEast}, what will be the belief state of the robot for its location? Show a belief state in this format:

$S = \{(r-x, c-y), (), ..., ()\}$ where $r-x$ is the row number, e.g. $r2$, and $c-y$ is the column number, $c3$ - according to the figure.

2. If the robot takes action "Right", and perceives P2={North, South, NoWest, NoEast}, what will its belief state S2 be for its location?

3. Assume that action "Right" makes a move to left, and action "Left" does a move to right. What will be the agent's belief state when it takes a "Right" action from its initial location?

7.1.1 Solution - Partially observable environment

1. Belief states on t=1

$S_1 = \{(r0, c0), (r3, c0), (r0, c7)\}$

2. Belief states on t=2, after action $Right$

$S_2 = \{(r3, c1)\}$

3. Belief states on t=2, after action $Right$ **that does** $Left$

The *initial location* refers to any of the belief state in question 1.

The robot doesn't move into/towards a wall, and hence the robot has the same perception and hence the same belief state as in the initial location.

$S = \{(r0, c0), (r3, c0), (r0, c7)\}$

Answers of the students who assumed the location depicted in the figure (r2,c3) as the initial location is also accepted. If this is the case, then the percept (**after moving left through a** *Right* **action**) is $P = \{NoNorth, NoSouth, West, NoEast\}$. The belief state is:

$S_{1B} = \{(r1, c8), (r2, c2)\}$

8 Planning (5 pts total)

8.1 Planning

You are developing a system that helps people make a plan to buy last minute Christmas gifts.

The input of the system is a specific gift a user wants to buy, and the output is a plan of how to buy it. Since there is no time to wait for delivery from an online store, the user would actually need to go to a physical store.

The following actions are available and can be used in the plan:

- *TakeBus* FROM somewhere TO somewhere else
- *DriveCar* FROM somewhere TO somewhere else, given that a USER *Owns* a specific CAR
- *Buy* a specific **gift** *At* a specific **store**

The system database contains two **stores**: BOOKSTORE and TOYSTORE, where a user can get either a BOOK or a TOY as **gifts**, respectively.

1. Specify the problem using Planning Domain Definition Language (PDDL), considering the USER is initially At HOME, Owns a CAR and wants to get a TOY as a gift. The goal is that the USER is back *At* HOME and *Owns* the TOY (2%)
2. Solve the problem using backwards search. Show the first 3 steps in one of the branches, including actions and states before/after each action (3%)

8.1.1 Solution - Planning

Specify the problem using PDDL

- $Action(TakeBus(from, to)$,
  - $PRECOND : At(user, from)$,
  - $EFFECT : \neg At(user, from) \wedge At(user, to))$
- $Action(DriveCar(from, to, car)$,
  - $PRECOND : Owns(user, car) \wedge At(user, from) \wedge At(car, from)$,
  - $EFFECT : \neg At(user, from) \wedge \neg At(car, from) \wedge At(user, to) \wedge At(car, to))$

- $Action(Buy(user, gift, from),$
  - $PRECOND : At(user, from) \land At(gift, from),$
  - $EFFECT : Owns(user, gift))$
- $Init(At(User, Home) \land Owns(User, Car), At(Car, Home), At(Toy, ToyStore))$
- $Goal(At(User, Home) \land At(Car, Home) \land Owns(User, Toy))$

Backwards search to solve the problem

1. State: $Goal(At(User, Home) \land Owns(User, Toy))$ plus some *useless* information like $Owns(User, Car) \land At(Car, Home) \land At(Toy, Home)$ that we omit for convenience...
   - To reach this, we had to $DriveCar(ToyStore, Home)$
2. State: $At(User, ToyStore) \land At(Car, ToyStore) \land At(Toy, ToyStore) \land Owns(User, Toy)$
   - To reach this, we had to $Buy(User, Toy, ToyStore)$
3. State: $At(User, ToyStore) \land At(Car, ToyStore) \land At(Toy, ToyStore)$
   - To reach this, we had to $DriveCar(Home, ToyStore, Car)$
4. State: $At(User, Home) \land At(Car, Home) \land At(Toy, ToyStore)$

So the Plan P looks like:
$P = \big[\ldots, DriveCar(Home, ToyStore, Car), Buy(User, Toy, ToyStore),$
$DriveCar(ToyStore, Home)\big]$ $\square$