NOTE: The workload in this exam was too high and I did some adaptations/adjustments in grading so that the final grades were comparable with the previous years' grades. Pinar, November 2021

QUESTION – SEARCH (25 points)

In the year 2100 humanity has branched out to many star systems in the Milky Way Galaxy. The decennial (10-yearly) human conference is being held soon, and you have been chosen as the ambassador of your star system. You are in star system S and have to travel to star system G, where the meeting is being held (strangely, we still haven't figured out how to remotely attend meetings despite being an interstellar species). The map of the nearby star systems are shown in Figure 1.

a) This task is about problem formulation. You are expected to formulate the problems described in the task using the "Well-defined problems and solutions" approach described in Chapter 3 in Russell and Norvig. Note that you are NOT expected to find a solution to the problem in a), you only need to write up the problem formulation.

Represent the following problem using the aforementioned method:
This is your first time representing your planet as an ambassador and you don't want to show up late, so you need to find the fastest route to get there, i.e the one taking the shortest amount of time. You therefore formulate the task as a search problem. Fill out the rest of the problem formulation by specifying the States, Actions, Path cost, Initial State, Transition model and Goal test such that the problem formulation can be used with A* to find the quickest path.

- Initial_state:
- States:
- Actions:
- Path_cost:
- Transition_model:
- Goal_test:

b) A* algorithm: Given the heuristic values presented in Figure 1, find the least costly path to star system G using the graph version of the A* algorithm. Tie breaks are solved in alphabetical order. Show the Frontier and Explored list in each step, as a table. Note that this question does not depend on your answer to a).

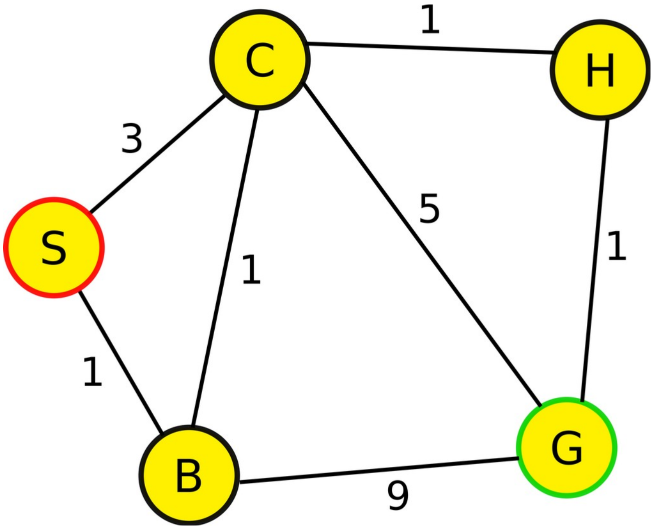


*Figure 1. Figure for A**

H(s) = 5
H(b) = 4
H(c) = 1
H(h) = 1
H(g) = 0

c) A* algorithm: Is the path you found above the optimal route to star system G? If it is not optimal can you make it optimal by changing one or more heuristic values? If it is not possible explain why, if it is possible show how.

d) Minimax algorithm: When you arrive at star system G you take out your speech to look it over one last time, but then an unknown life form grabs your speech and runs off with it. You chase after the life form. Fortunately, one of the organizers saw what happened and helps you. Eventually, you are able to corner the life form in a room as shown in Figure 2. The organizer knows that this particular type of alien loves "bluuurgh", a spiky type of food. In the figure, Y is you, O is the organizer, the green thing is the life form, and the blue one is food.
Now a turn-based game will be played that has the following rules:

1. The life form moves either Left, Right, Up of Down.
2. You move either Up or Right
3. The organizer moves either Down or Left.
4. Everyone must make a move.

If the alien life form enters the square with the bluuurgh in it, it gets a reward of +10. If either you or the organizer enters the same square as the alien life form you will then catch it and get the speech, giving the alien life form a reward of -30. The game ends when you've either caught the alien life form or everyone (i.e., each of you, organiser and the life form) has made one move. Construct a minimax tree for the situation described with this order of players: Alien Life form, You, and

Organizer. The alien life form is the maximizer and you and the organizer are both minimizers.

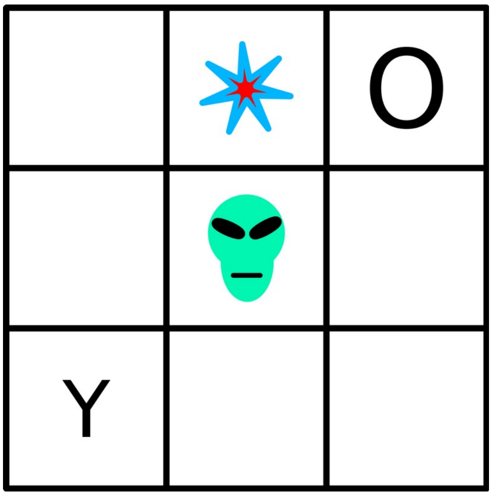


Figure 2. Figure for Minimax alg

e) Alpha-Beta pruning: Use minimax algorithm with alpha-beta pruning on the tree given in Figure 3. Successor nodes are examined from left to right.

   a. Which nodes, if any, are pruned?
   b. Which move ordering prunes the most nodes?
      Def: Move ordering is the order that successor nodes are examined. Changing the move ordering does *not* change the parent-child relationships between any nodes.

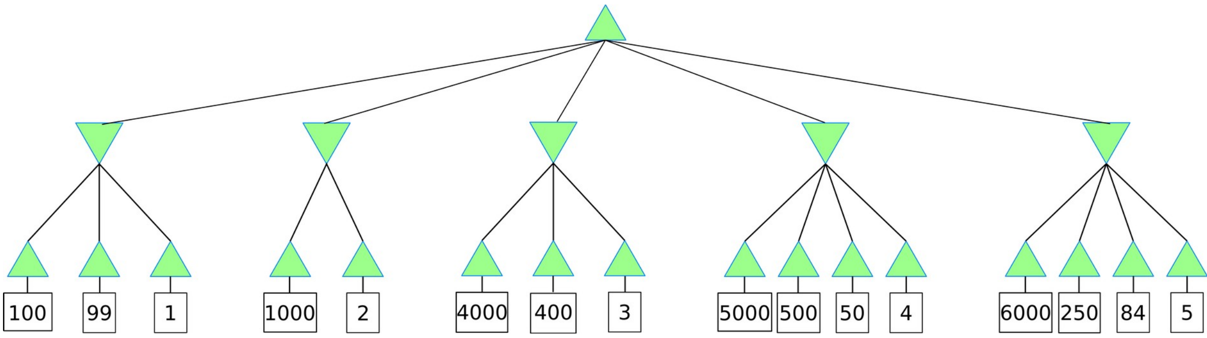


Figure 3. Figure for Alpha-Beta pruning

f) Breadth-first search: Having arrived in the meeting star system you must now find the planet where the meeting is being held. Use breadth first graph search to find a path from the starting planet S to the meeting planet G and show the node expansion order. Tie breaks are solved alphabetically. Write down the path found by breadth first graph search. Is this the optimal path?

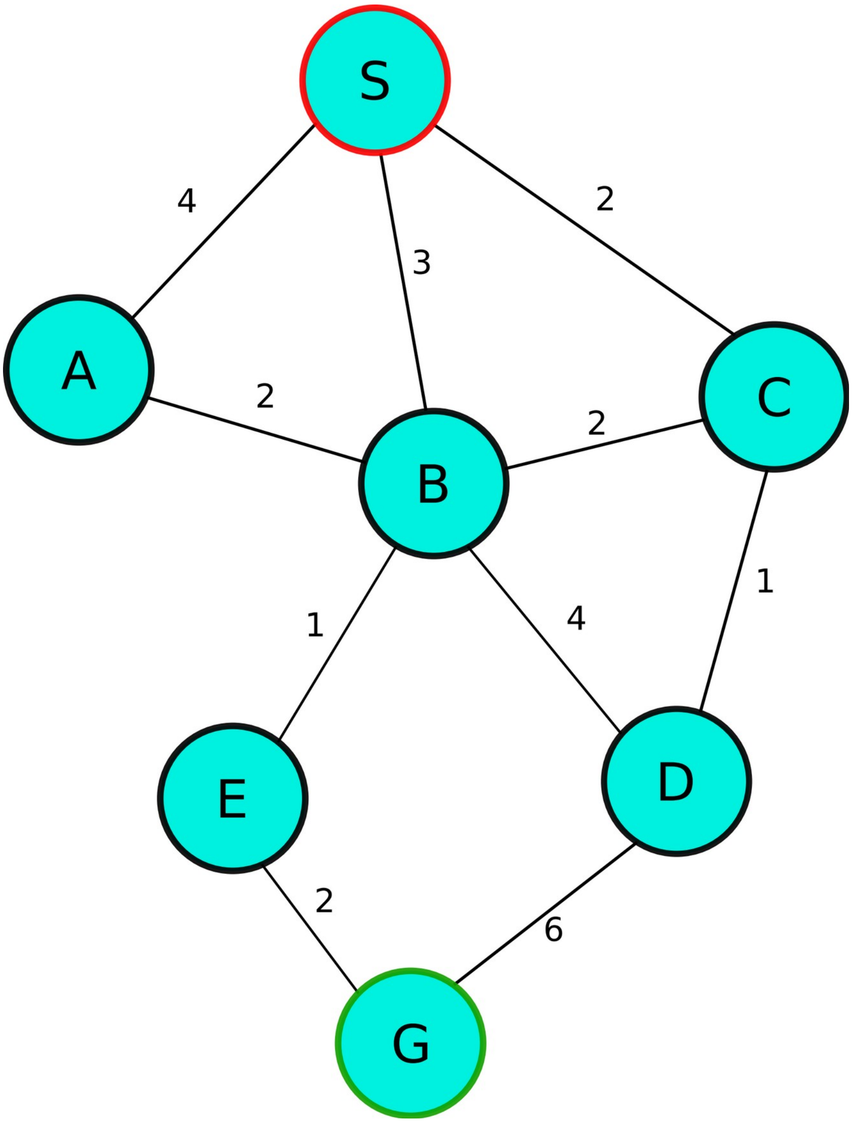


Figure 4. Figure fir Breadth-first search

QUESTION – LOGIC (total 25 points)

In this exercise you will be solving tasks related to a Cave Spelunking game on a 2D board, shown in Figure 5. In this game every square on the 2D board can contain Lava, Diamonds, one and only one Player, Sparkles, and Red Glow or nothing. To win the game the player must find every diamond on the board by moving to every Diamond square. If you move into any lava square you lose the game. Every square that is vertically or horizontally adjacent to a lava square has a Red Glow, while every square that is vertically or horizontally adjacent to a diamond square Sparkles. The Player starts at a specific location in each instance of the game.

Please note that although these problems are thematically similar they can all be solved independently of each other, that is, the answer to one question never depends on the answer to another question.

a) Your first task is to represent the following board state in either propositional or predicate logic. You should pick whichever logic you prefer and comment on your choice, explaining why you prefer one logic over the other for this use case – the question is not looking for a specific answer, but rather for you to show that you are able to reflect and justify your decision. Please specify the semantic meaning of the symbols you use.

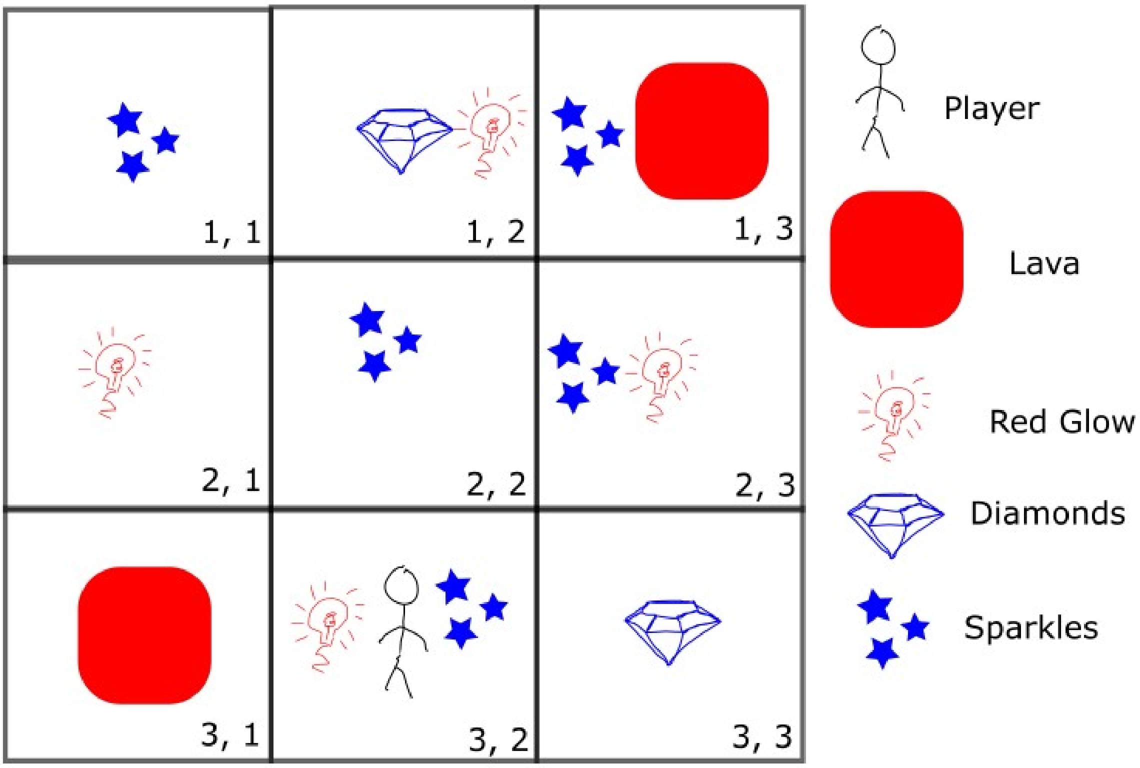


Figure 5. Figure for Logic quest (first task)

b) The following is an (incomplete) set of rules for the Cave Spelunking game expressed in natural language. Express them using predicate logic and explain the semantic meaning of the symbols you use if it is not obvious (i.e. if you called something *Cactus(x)* you would have to explain what this means). You do not need to consider temporality; the question wants you to represent the sentences in predicate logic and not in a logic where they may be applied sequentially.

1. Every square adjacent to a diamond sparkles
2. Every square adjacent to lava has a red glow
3. If the player is in the same square as lava the game is over
4. If the player finds every diamond in the game the player wins – that is, there is no squares left with any diamonds
5. If the player is in the same square as a diamond the player has the diamond
6. The player can only move to a square adjacent to a square that has Red Glow if the square is also adjacent to a square that has Sparkles.
7. The player can only move to squares adjacent to the square the player is currently in.

c) Given the rules of the Cave Spelunking game as specified in (b), is it possible for the player to win given the board Figure 6. Explain why/why not. You may assume the player always takes the best action it is allowed to given that the they have to follow the rules in (b). Please also ignore the fact that propositional logic is not a temporal logic and that it is a monotonic logic.

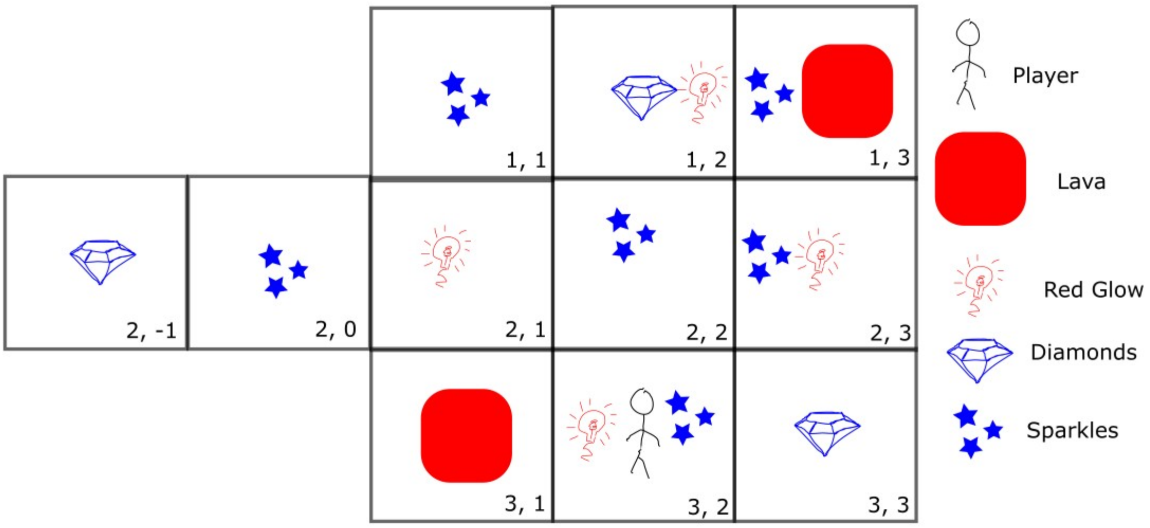


Figure 6. Figure for Logic Quest, Task c.

d) The following problem does not use any rules mentioned earlier. Given the following sentences in propositional logic, translate the sentences to CNF and use resolution refutation to prove the sentence “Richard does not mine diamonds”. When you convert the sentences to CNF explain every step you do.

Rules:

a. $Afraid(Richard, Lava)$

b. $Person(Richard)$

c. $\forall x \forall y (Person(x) \land Afraid(x, y) \Rightarrow \neg Close(x, y))$

d. $Close(Diamonds, Lava)$

e. $\forall x \forall y \forall z (Close(x, y) \land Close(y, z) \Rightarrow Close(x, z))$

f. $\forall x \forall y (Close(x, y) \Rightarrow Close(y, x))$

g. $\forall x \left(CanMine\left(x, Diamonds \right) \Rightarrow Close\left(x, Diamonds \right) \right)$

Intended interpretation:
Richard= A name of a human
Lava= Very hot rock, deadly
Diamonds= Precious gemstone
Afraid(x, y)= x is afraid of y
Close(x, y)= x is close to y
CanMine(x, y)= x can mine y
Person(x)= x is person

e) For each of the following sentences, write whether the sentence is valid, satisfiable, or neither. Comment/explain on your answer with one sentence (you do not need a detailed explanation, just make it clear how you reasoned).

a. RG
b. $RG \wedge \neg RG$
c. $RG \vee \neg RG$
d. $RG \Rightarrow RG$
e. $RG \Longleftrightarrow D$
f. $\neg \left(RG \Longleftrightarrow D \right)$
g. $\left(\left(RG \vee D \right) \wedge \neg \left(RG \wedge D \right) \right) \Rightarrow RG$

QUESTION - CONSTRAINT SATISFACTION

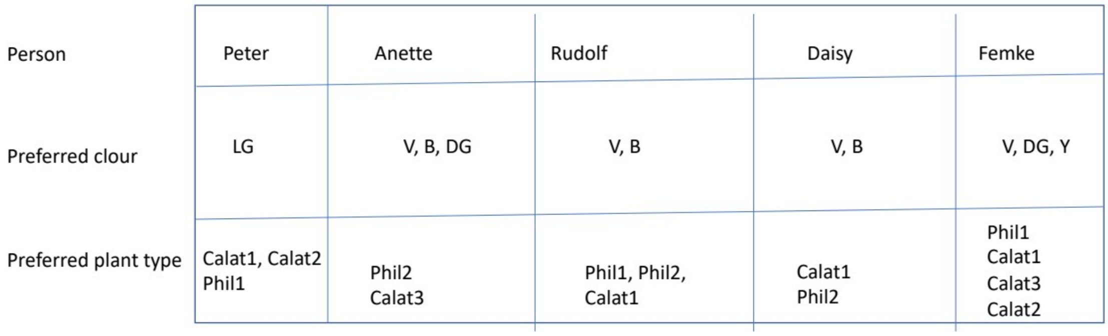
Assume five classmates, Rudolf, Anette, Peter, Daisy and Femke, from NTNU has moved to Oslo and as part of their house decoration each bought a plant. It is also known that the plants are either in the Philodendron or the Calathea category. You also know that the plants the classmates own are one of these: Phil1 or Phil2 from the Philodendron category, or Calat1, Calat2, or Calat3 in the Calathea category. We also know that the colour of the leaves of the plants can be light green(LG), dark green(DG), yellow(Y), blue(B), and variegated(i.e., more than one colour,V).

Each person has a different colour and a different plant than all the other 4 friends. For example if one has a LG coloured plant, none of the others can have a LG coloured plant. Similarly. if one has a Calat2 plant, then none of the others can have a Calat2.

The CSP problem: You want to find out which person may have what kind of plant, i.e., which plant and its colour.

You know somethings about the favorites of these people, which is shown in Table 1, which you must take into consideration when solving some of the following tasks.

Table 1. CSP problem,domains



Person	Peter	Anette	Rudolf	Daisy	Femke
Preferred clour	LG	V, B, DG	V, B	V, B	V, DG, Y
Preferred plant type	Calat1, Calat2 Phil1	Phil2 Calat3	Phil1, Phil2, Calat1	Calat1 Phil2	Phil1 Calat1 Calat3 Calat2

In addition, you know that:

- Those persons who have Blue as favourite color and Phil2 as their favourite plant do not want any other blue coloured plant or any Phil2 with any other colour.
- Daisy and Peter have a plant in the same plant category (i.e, both have either Philodendron or Calathea)
- Anette and Femke also have a plant in the same plant category.

1. After learning that “Those persons who have Blue as their favourite color and Phil2 as their favourite plant don’t want any other blue coloured plant and any Phil2 with other colour”, you can reduce the domains of some people immediately. Write down the domains for all variables taking into consideration this information. Use the notation where a value in a domain is a pair of color and plant type (eg. LG-Calat1).
2. Draw the **search tree** (complete the one below) that results from applying **backtracking search with forward checking**. Don’t do&show the complete search tree but only for the assignment of LG-Clalat1 and LG-calat2 for Peter.
   Use the reduced domains you made in task 1 and assign the variables in the order shown in the tree below (i.e., Peter, Anettte, Rudolf, Daisy, Femke). For each variable assign the values like this: first color and first plant type (i.e., pair of colour and the plant type) , then the first colour and the second plant type, etc. where the order of the colours is **LG, V, B, DG, Y**, and the order of plants is Calat1, Calat2, Calat3, Phil1, Phil2. Complete the search tree in the figure 7.

3. Draw the search tree that results from applying **backtracking search with forward checking and propagating the check through domains that are reduced to singleton domains.** That is, if a domain with a single value appears during the consistency

check after assignment of a variable, then the check propagates along this domain for all unassigned variables. Use the domains found above, in 1. Draw the search tree STARTING WITH FIGURE BELOW Figure 7.

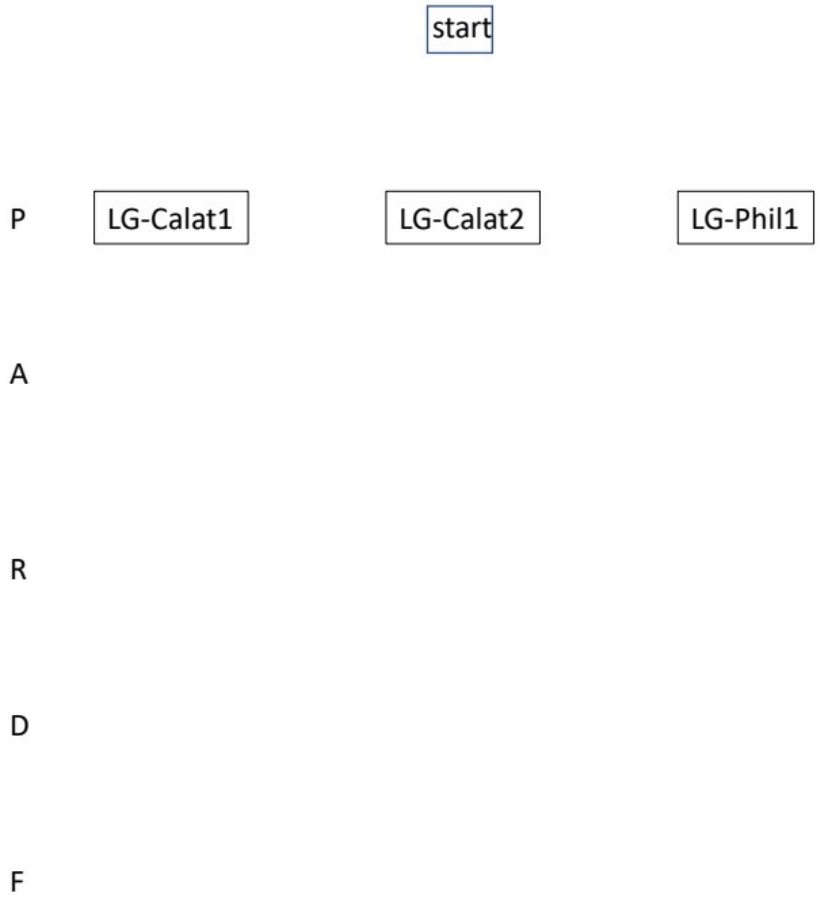


Figure 7. CSP problem. Compete this serach tree

QUESTION –PLANNING

1. An agent has a task of painting two chairs, A and B, and stacking A on top of B. A Chair is “free” (i.e., Free(A)) if there is nothing on it preventing it being painted

(e.g., nothing is on top of it). The agent has three types of actions, *sand*, *paint*, and *putOn*:

sand (x)

precondition: Free(x)
postcondition: Smooth(x)

paint(x)

precondition: Smooth(x), Free(x)
postcondition: Painted(x)

putOn(x,y)

precondition: Free(x), Free(y)
postcondition: On(x,y), $\neg Free(x), \neg Free(y)$

In Figure 8 you see a partially-ordered plan that is not correct because it has some problems/flaws.

a) Give at least 3 reasons why it is not correct. In case an action is the cause of more than one flaw, write each one as a separate flaw and number them as Flaw 1, Flaw 2, …… in a list
b) For each flaw write the plan operation that is needed to resolve the flaw.
c) Write down the final plan a semicolon between the actions in the plan where the action on the left of a semicolon will be executed before the one on its right. When the order of two actions are not important, write them within a parenthesis with an "AND" between them, e.g., (action-x AND action-y). Hence a plan may look like this:

action-z ; (action-x AND action-y) ; action-m.

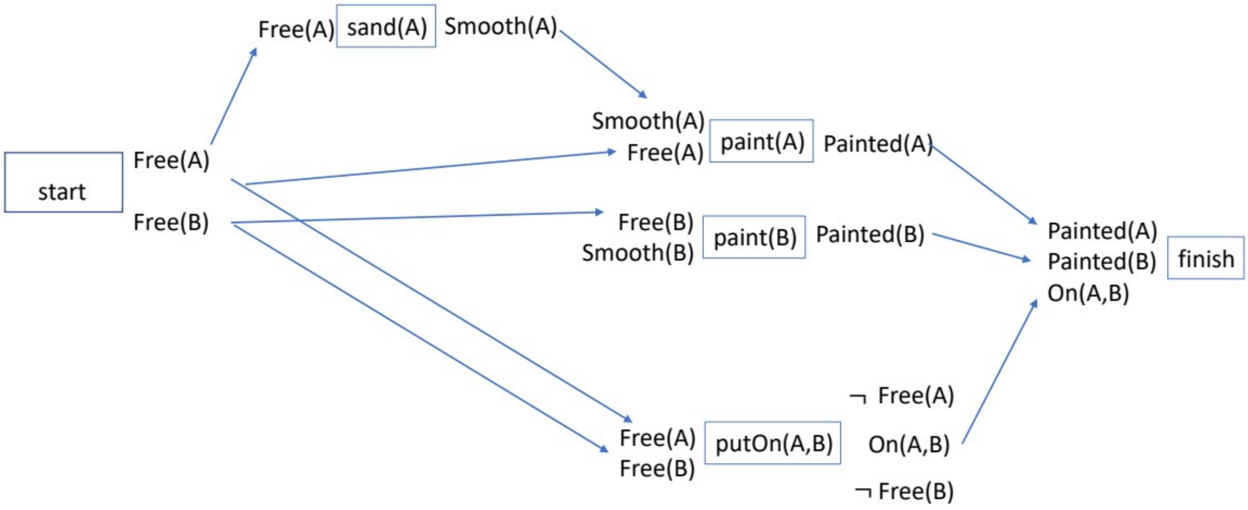


Figure 8. Figure for planning

2. This question is independent from the preceding one. Assume that the GraphPlan algorithm successfully generates a plan in one action layer only, and there are 3 actions in that plan. What kind of constraints may be there regarding the order of the execution of these actions?

3. This is a planning problem and is domain independent. In the given planning problem the goal is: $C \wedge D$
   There are two action schemas, specified below.

   Action: tik
   Precondition: none
   Postcondition: C, D

   Action: tak
   Precondition: none
   Postcondition: $D, \neg A$

   a) Find the solution that Partial-order planner returns, and write it down. If there are more than one solution write down each as an alternative plan. Show the planning process on a figure.

   b) Find the solution that the GraphPlan algorithm would return, and write it down. If there are more than one solutions write down each as an "alternative plan". Show the planning process on a figure.

   c) What is the main reason for the two planning methods' behaving differently in this particular planning problem/case.

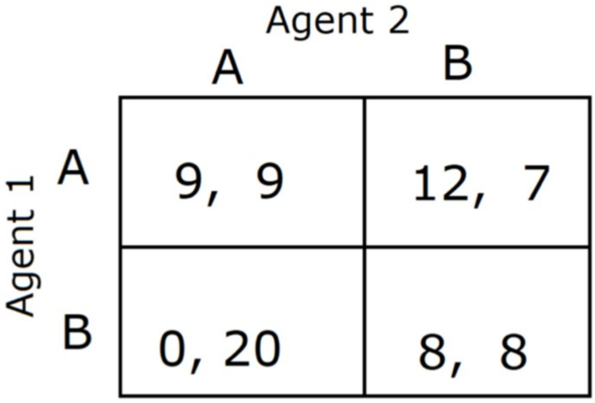
QUESTION - GAME THEORY

1) Fifty students taking TDT4136 are asked to choose an integer between zero and hundred. They will play a game where the payoffs are calculated like this: If a

number 0£ k £ 100 is chosen strictly more times than all other numbers, then the students who chose k will get a Freia Milk Chocolate plate while others get nothing. If no number was chosen strictly more times than any other number, nobody gets chocolate.

a) How will you represent this story in Game Theory? Since you cannot show the payoffs in a matrix, use a function to represent utilities.
b) What is the Nash equilibrium (of equilibria if more than one) of this game if there is any? Explain your answer/reasoning.

2) A game with two agents has the payoff matrix shown in Figure 9.

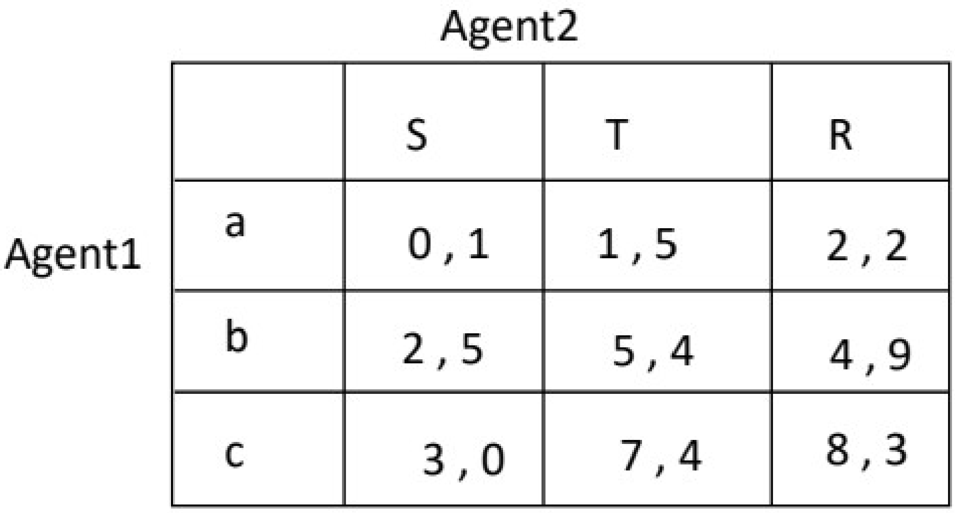


		Agent 2	
		A	B
Agent 1	A	9, 9	12, 7
	B	0, 20	8, 8

Figure 9. GT. Task 2, Find pareto etc

a) Find the pareto optimal strategy of the game. Write down all if there are more than one.
b) Write down the social optimum strategy. Write down all if there are more than one.

3) Two agents are playing a game where the payoff matrix is shown in Figure 10:



		Agent2		
		S	T	R
Agent1	a	0 , 1	1 , 5	2 , 2
	b	2 , 5	5 , 4	4 , 9
	c	3 , 0	7 , 4	8 , 3

Figure 10. GT. Fig for Elimination of dominated strategies

Find the solution of the game through eliminating dominated strategies. Describe every step of elimination.

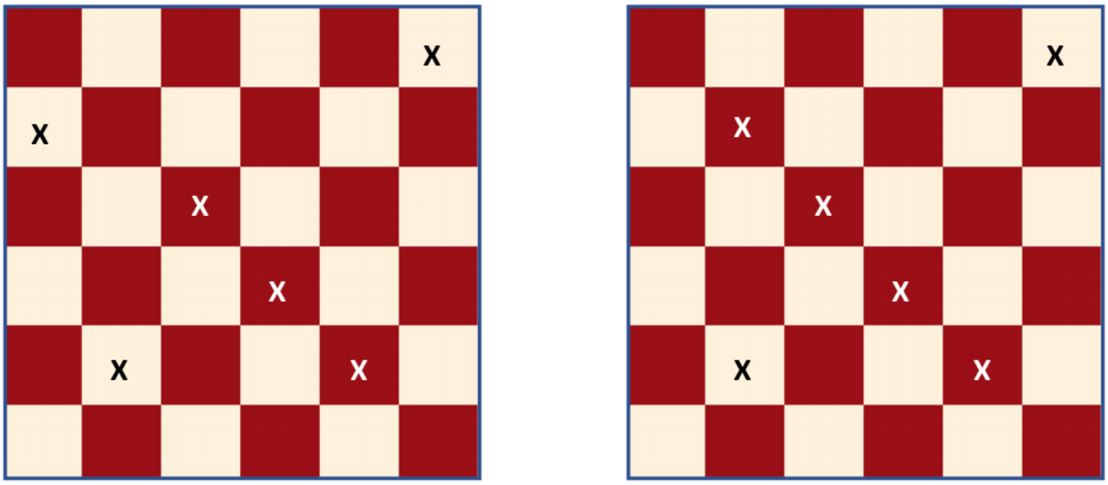
QUESTION - SHORT QUESTIONS

1. **Local search**

   The goal of the 6-queen problem is to place 6 queens on a chessboard such that no queen attacks any other. A queen attacks any piece in the same row, column or diagonal. We define the rules of this puzzle as follows:
   A state consists of a queen in each column. A queen can be moved to another square in the same row. The evaluation function is
   Eval(s) = 1- # attacking pairs in state s.
   The following figure illustrates two states: the current state (a in Figure 11), and a state that the basic Simulated Annealing algorithm generates randomly as a candidate next state. You need to find out what the algorithm will decide regarding to move or not to the next randomly generated state (b in the figure) from the current state (a in the figure).

   a) Which values will the Eval function return in the current situation shown in (a) in the figure, and the next state candidate in (b) – i.e., Eval(a) and Eval(b)? Explain how you found the numbers.

b) With which probability will the SA algorithm accept this move to (b)? Explain your answer and provide a number for the probability, between 0 and 1. Assume the temperature T=4.



(a) Current state
(with 5 attacking pairs)

(b) Randomly generated possible next state

6-Queens problem

Figure 11. Figure for local search

2. Ethical Issues

One of the approaches to ethics we have seen is the consequentialist approach. How can you connect this approaches to the rest of the syllabus in this course? Explain your reasoning.

3. PEAS and Characteristics of the Environment

Consider a problem domain where an autonomous boat is dropping stone blocks into the water to safely detonate hidden mines. Give a PEAS description of this task environment and characterize it in terms of the properties of task environments as defined in Chapter 2 in Russel and Norvig.

4. Translation from Semantic Networks to Logic

Translate the sematic network representations shown in Figure 12 to logic representations. Note that an instance is like a member. For example, Clyde is-a (or a member-of) Elephant class/category.

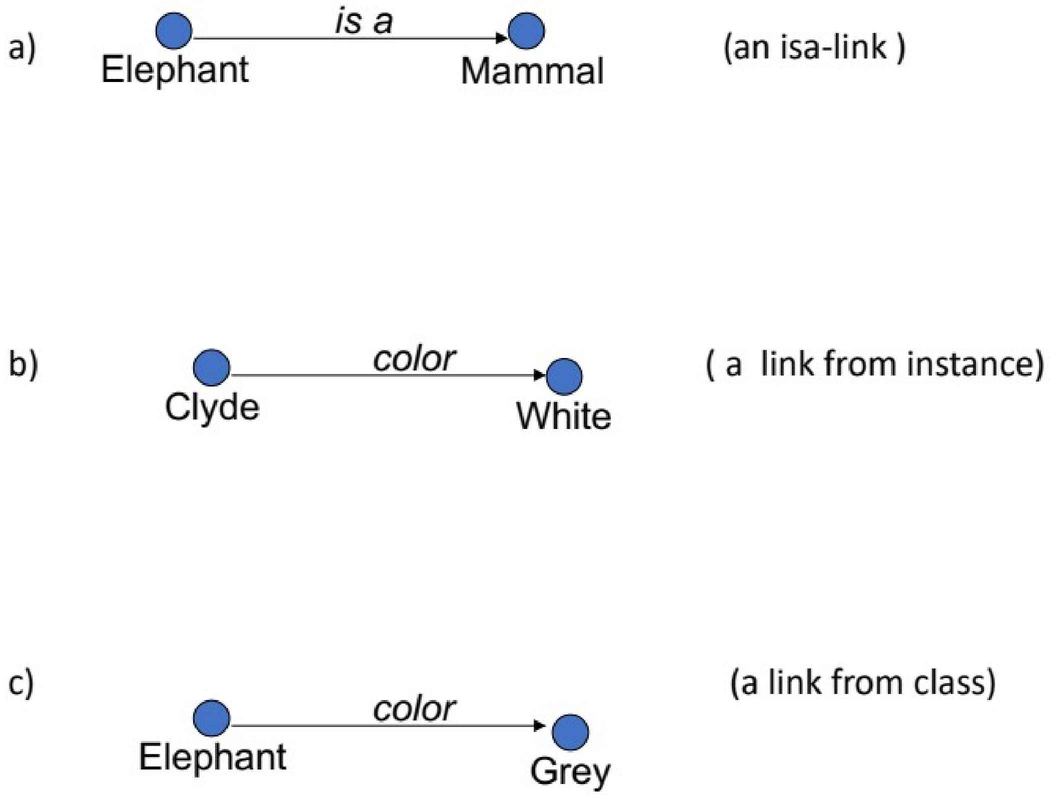


Figure 12. Translation from SN to Logic