## OPPGAVE 1 (True/False questions) (20 pts, 2 pts for each correct answer, -1 for each wrong answer. Total points will not be less than zero) *Mark each of the following sentences either as True or False*

1. True or False: Greedy best-first search algorithm is guaranteed to find an optimal path.
2. True or False: Uniform Cost algorithm is guaranteed to find the optimal solution when all step costs are greater than zero and the branching factor is finite.
3. True or False: Let *b* be the branching factor of a search, *d* the depth of the solution, and m the maximum depth of the search space. Then the complexity of breadth-first search is $b^m$.
4. True or false: $(A \Leftrightarrow B) \vDash (\neg A \vee B)$
5. Linear planning is incomplete.
6. Simple "hill climbing" algorithm is perfect to solve constraint satisfaction problems.
7. A sound logical reasoning process is not necessary in order to pass the Turing test.
8. Depth-first tree search algorithm always expands at least as many nodes as an A* tree search algorithm with admissible heuristic does.
9. The set consisting of "mammal" and "non-mammal" categories is both a disjunctive and an exhaustive decomposition of the category "animal".
10. Explainability is an ethical problem in AI which domain knowledge may help to solve.

## OPPGAVE 2 (First Order Logic) (15pts – 7-3-5)

The Knowledge base has the following sentences for which FOL representations are given below.

Everyone who loves all animals is loved by someone.
Anyone who kills an animal is loved by no one.
Sofie loves all animals.

Either Sofie or CarAccident killed the cat, who is named Kismet.

*FOL representations:*

1. $\forall x\ [\forall y\ [Animal(y) \Rightarrow Loves(x,y)]] \Rightarrow [\exists z\ Loves(z,x)]$
2. $\forall x\ [\exists y\ (Animal(y) \land Kills(x,y)] \Rightarrow \neg(\exists z\ Loves(z,x))$
3. $\forall x\ [Animal(x) \Rightarrow Loves(Sofie, x)]$
4. $Kills(Sofie,Kismet) \lor Kills(CarAccident,Kismet)$
5. $Cat(Kismet)$

*Query*: Did CarAccident kill the cat?

You are going to the answer the above query by using resolution by refutation.

1. First convert all FOL sentences into conjunctive normal form (CNF). Show every step in this process.
2. Write down any background knowledge that is needed to solve the problem
3. Apply resolution by refutation and show how the query is answered. Show each and every unification.

## OPPGAVE 3 - A* search algorithm (13 pts, 1-4-4-4)

1. Apply A* algorithm for graphs on the graph in the following figure (Figure 1). Write down the nodes in the order they are expanded. A is the start node, and G is the goal node.
2. What is the returned path? Is it optimal? explain your answer on an example from the given graph in the figure.
3. Modify the pseudocode in the figure (Figure 2) for A* algorithm for graph search so that it guarantees to find the optimal solution with heuristic values given in Figure 1. Write down the pseudocode in a separate paper starting from the sentence just before your modification starts, and ending with the sentence right after your last change. That is, you don't need to write the whole pseudocode, only the part you modified, plus a single/one original sentence before and one after your modified sentences.
4. Assume that a search tree has heuristic values which enable the A* algorithm presented in the pseudocode (the unmodified version in figure 2) to find an optimal path to the goal. Would the A* algorithm still be guaranteed to find the minimal path to the goal if there are negative transition costs? This question is general, not about the problem presented in Figure 1.

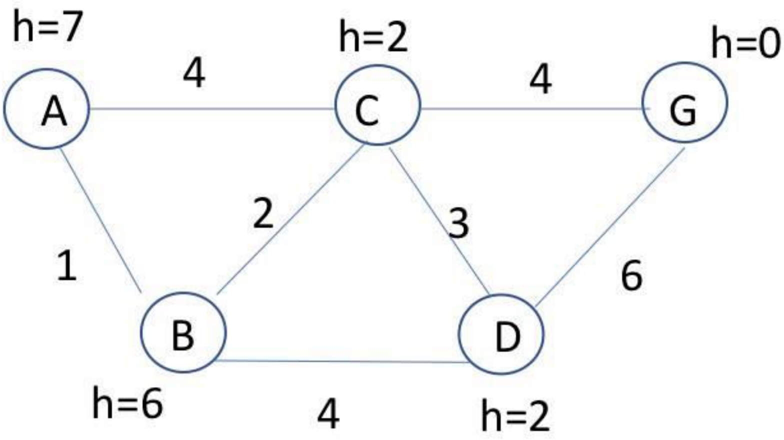


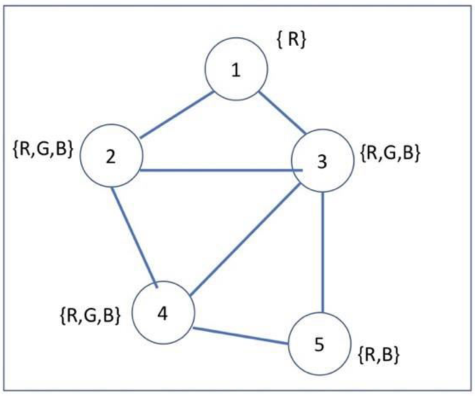

*Figure 1 Graph for the question. A is the start node, and G is the goal node.*

```
Start.g = 0;
Start.h = heuristic(Start)
FRONTIER = {Start}
CLOSED = {empty set}
  WHILE FRONTIER is not empty
      N = FRONTIER.popLowestF()
      IF state of N= GOAL RETURN N
      add  N to CLOSED
      FOR all children M of N not in CLOSED:
          M.parent = N
          M.g = N.g + cost(N,M)
          M.h = heuristic(M)
          add M to FRONTIER
      ENDFOR
  ENDWHILE
```

*Figure 2 .Pseudocode for A* algorithm.*

## OPPGAVE 4 - CONSTRAINT SATISFACTION (12 pts, 2-4-2-4)

Consider the following constraint graph (in the figure) for a graph coloring problem where the constraints mean that the connected nodes cannot have the same color. The variables are shown inside the nodes while the domains are shown next to each variable node.




1. What are the domains after a full constraint propagation using an arc consistency algorithm?
2. Show the sequence of variable assignments during a pure backtracking search (don't assume that propagation above has been done). Assume that the variables are examined in numerical order and the values are assigned in the order shown next to each node. Show assignments by writing the variable number and the letter for the value, e.g, 5R, 2G.
3. Describe how forward checking works.

4. This time you'll apply backtracking search with forward checking. Use the same ordering convention for variables and values as above. Show the sequence of variable assignments during backward search with forward checking. Again, show assignments by writing the number of variables followed by the letter for the value.

## OPPGAVE 5 – Game Theory (10 pts, equal points for each question))

Consider the game for which the payoff matrix is shown in the following figure.

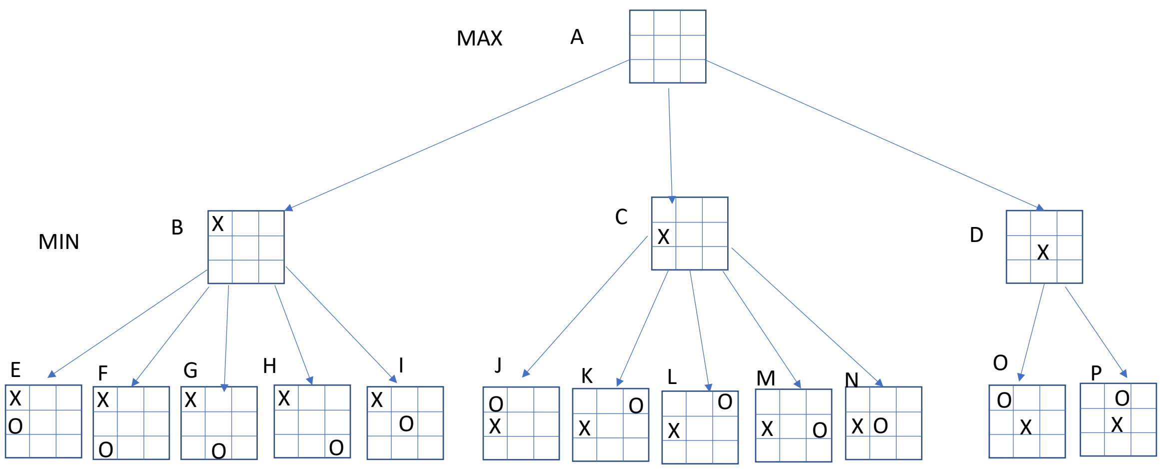
| | | agent2 | |
|---|---|---|---|
| | | G | NG |
| agent1 | H | 8, 0 | 3, 1 |
| | NH | 4,4 | 2,3 |

*Figure 3 Payoff Matrix.*

1. Identify any *dominated* strategy. Explain your answer.
2. Find the *Nash equilibrium*. What are the equilibrium payoffs, i.e., values for each agent?
3. Are there any *pareto optimal* joint actions? If any exists, what are they?
4. Explain (in general, not for this particular problem) why a *social welfare maximizing* joint action profile is also pareto optimal.

## OPPGAVE 6 - ADVERSARIAL SEARCH (10 pts, 3-3-4)

Consider a tic-tac-toe game on a 3x3 grid (see the figure below) where players MAX and MIN take turns marking the spaces in a 3×3 grid by placing their X's and O's, respectively. The player who succeeds in placing three of their marks in a horizontal, vertical, or diagonal row is the winner.



The values of the terminal nodes have not been provided. You will compute these values using an evaluation function *e*. Function *e* uses a heuristic that estimates the value of each terminal node according to the following formula:

e(node)= E1-E2 where

E1=sum of the number of rows, columns and diagonals that are possible winning situations for Max and,

E2= sum of the number of rows, columns and diagonals that are possible winning situations for Min.

The following figure shows examples for computing the values of some hypothetical nodes:

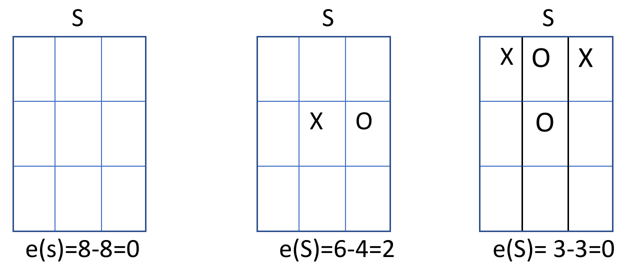



*Figure : Example on estimation the values of terminal nodes*

1. Compute and write down the values of each terminal node node, i.,e of E, ……, P.
2. Using the *Minimax* algorithm find and write down the values of the remaining nodes(A,…D) in the search tree. Which action will Max play in this game, the action whose outcome is B, C or D?

3. Is it possible to prune any nodes using the *Alpha-Beta pruning* algorithm? If there are any, write down the prunable node(s) - i.e., write the letter for the node.

## OPPGAVE 7 - SHORT QUESTIONS (20 pts, 2 pts for each question. Plus 2 points for correct answer to the "bonus"/voluntary question.)

1. How is the goal information represented in simple reflex agents?

2. Is the PL sentence $((\mathrm{P} \rightarrow Q) \wedge Q) \rightarrow P$ valid, unsatisfiable or satisfiable? Justify your answer.

3. Translate the following sentence into predicate logic:
"Any person who has an umbrella is not wet"

4. Translate the following sentence into predicate logic:
"John has at least two daughters."

5. Assume a Hill Climbing algorithm that aims to find the best state according to a heuristic cost function. Does it try to find the global minimum or the global maximum?

6. It has been suggested that the first phase of *GraphPl0an* be used as a heuristic function for forward search in the following way: Given a state *s* and goal *g*, run the graph-construction phases of *GraphPlan* until all the components are represented and not mutex in the last layer. Let *n* be the number of action layers in the graph. We will let *n* be the heuristic value for s. Is this an admissible heuristic? Explain your answer.

7. If *GraphPlan* terminates with a successful, 3-action plan in the first iteration, what constraints are there on the order in which the actions must be executed?

8. Does *Regression Planning* work in a forward or backward manner?

9. You will represent the concept of "student" using a *frame-based* knowledge representation language. You want the age of a student to be computed on the basis of her birth year and the current year. How would you represent this in a slot of "student" frame?

10. Assume a version of the original vacuum cleaner agent in the textbook. 10% of the time the SUCK action of this one does not clean the floor if it is dirty and even may deposit dirt on the dirt on the floor if the floor is clean. Classify this environment with respect to each of the following dimensions:
    Sequential/Episodic, deterministic/stochastic, and dynamic versus static.

11. BONUS QUESTION: Assume that *Simulated Annealing* search algorithm starts from a state $S_0$ in the middle of a large plateau. That is, the values of all states on the plateau are exactly the same. Assume also that in the first step the random neighbor we picked is $S_1$, which has the same value as $S_0$. Will Simulated annealing move to $S_1$? Explain your answer mathematically (i.e. using a formula). No points will be given otherwise.