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:

a)

- Initial_state: The initial state is the starting star system S
- States: The states are the set of all states reachable from the initial. state by a sequence of actions.
- Actions: The actions are the possible action available in a state. Given a state s, ACTIONS(s) returns all actions that can be performed in s. For instance ACTIONS(C) = {Go(S); Go(B); Go(H); Go(G)}

• Path_cost: The path cost is the sum of costs of the individual actions along the path. They are shown on the graph. For instance COST(In(S);GO(B)) = 1

• Transition_model: Given a state and action, this returns the resulting planet: RESULT(In(B); Go(C)) = In(C)
• Goal_test: The goal test checks whether a given state is the goal state. Here the goal is the singleton set {In(G)}

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).

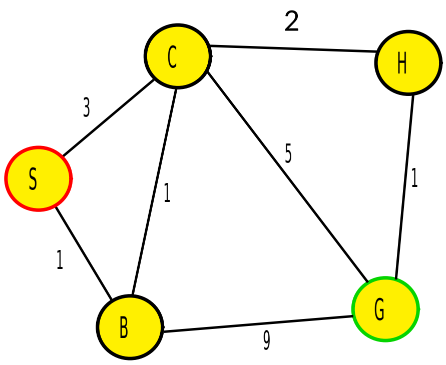


*Figure 1. Figure for A**

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

b)

Explored	Frontier
	S(5)
S(5)	C(4,SC), B(5,SB)
S(5), C(4, SC)	H(6, SCH), G(8, SCG), B(5, SB)
S(5), C(4, SC), B(5, SB)	H(6, SCH), G(8, SCG)
S(5), C(4, SC), B(5, SB), H(6, SCH)	~~G(8, SCG)~~, G(6, SCHG)
S(5), C(4, SC), B(5, SB), H(6, SCH), **G(6, SCHG)**	

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.

c)

• Path found: SCHG, cost = 6. Not optimal.
• Reason: Heuristic is inconsistent.
• Fix: Change e.g h(C) = 3, then A* will find SBCHG which has a cost of 5 and is optimal.

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.

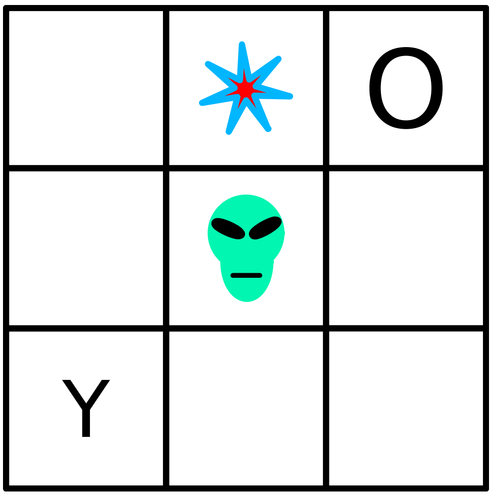
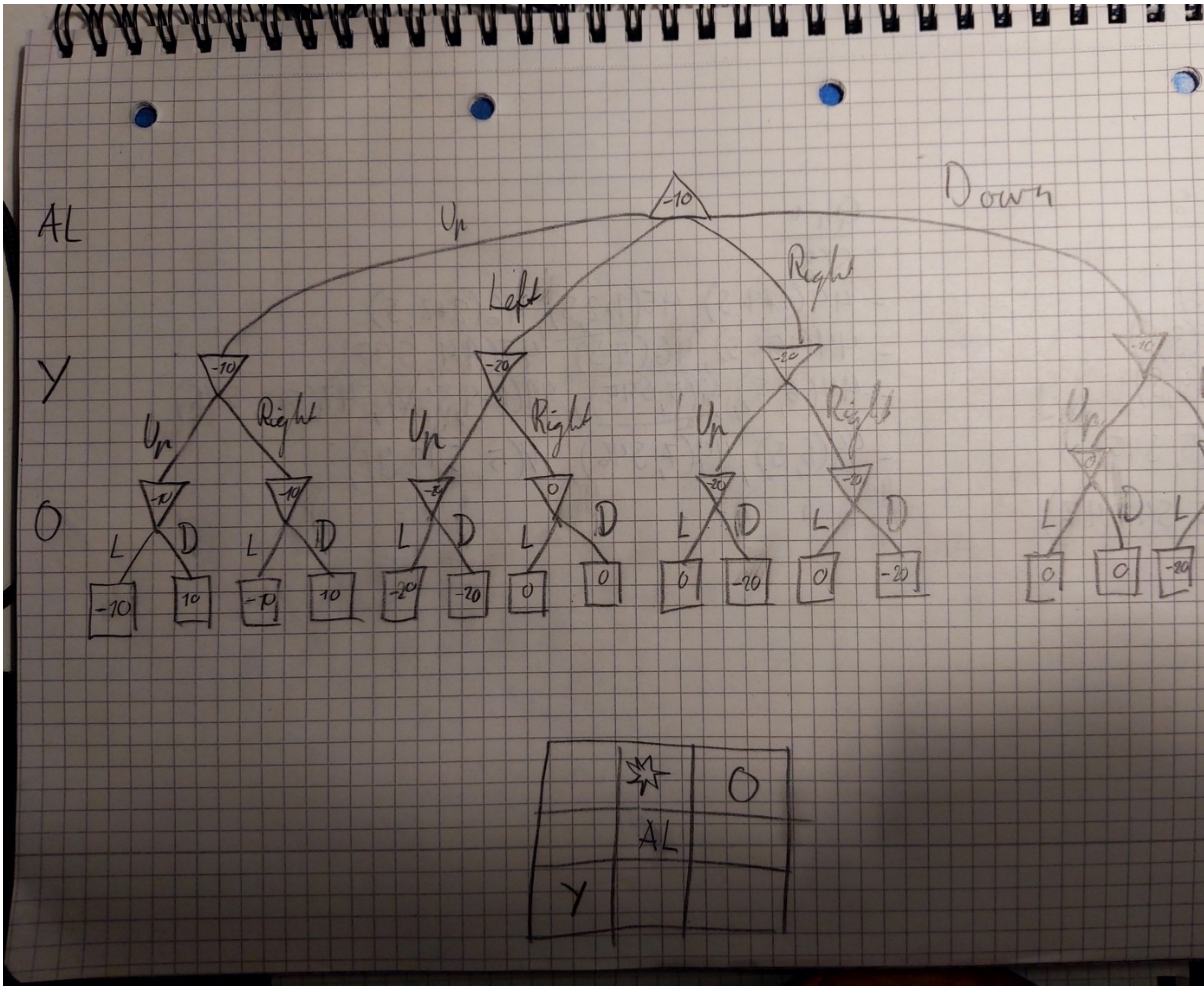


Figure 2. Figure for Minimax alg

d)



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.

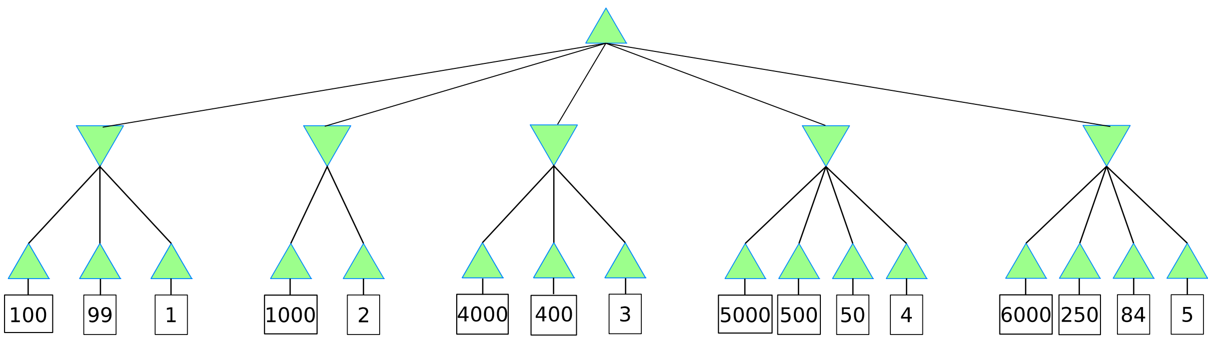


Figure 3. Figure for Alpha-Beta pruning

e). See Figure 2 below.
a: No nodes are pruned.
b: Search the tree from a right to left manner instead.
This corresponds to rearranging the nodes so that the three looks like. See figure 3

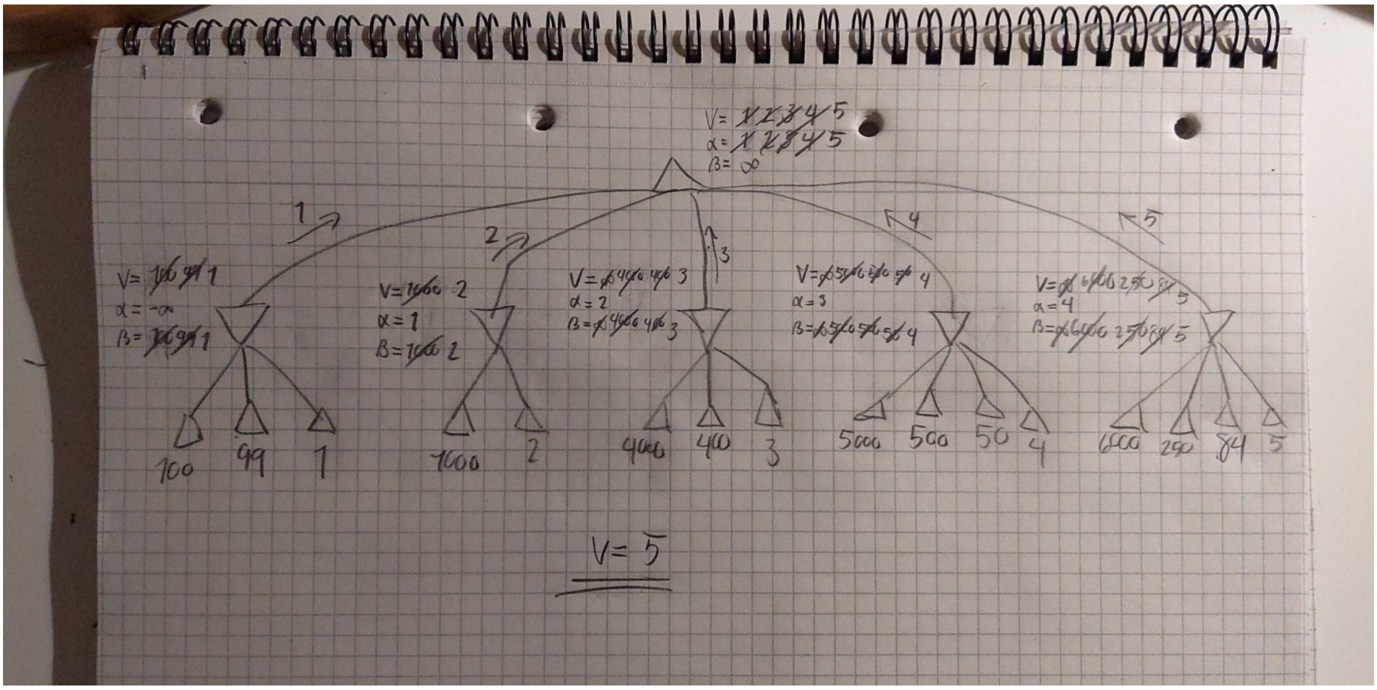


Figure 2

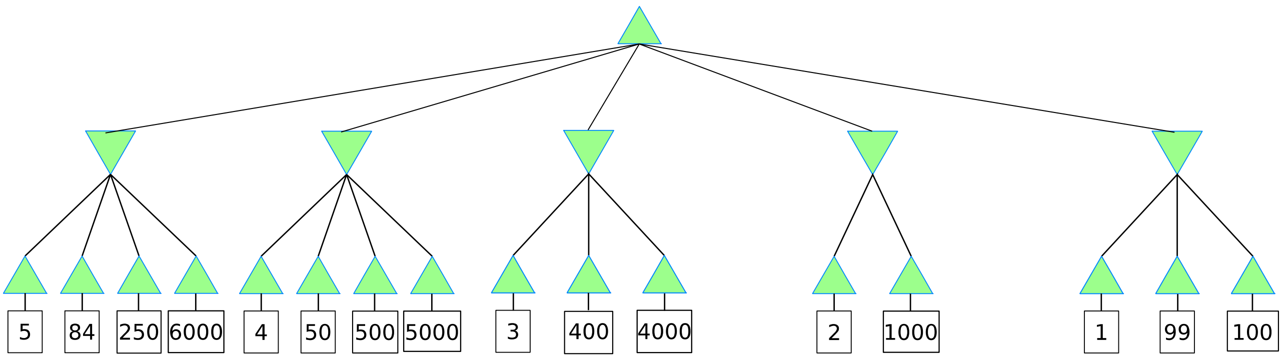


Figure 3

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?

f). **Found solution is not optimal, optimal solution would be SCDG**

Explored	Frontier	Current Node
	S	
S	A(S), B(S), C(S)	S
S, A(S)	B(S), C(S)	A(S)
S, A(S), B(S)	C(S), D(S, B), E(S, B)	B(S)
S, A(S), B(S), C(S)	D(S, B), E(S, B)	C(S)
S, A(S), B(S), C(S), D(S, B)	E(S, B), G(S, B, D)	D(S, B)

Path found: SBDG
It is NOT the optimal path.
The optimal path is: SBEG

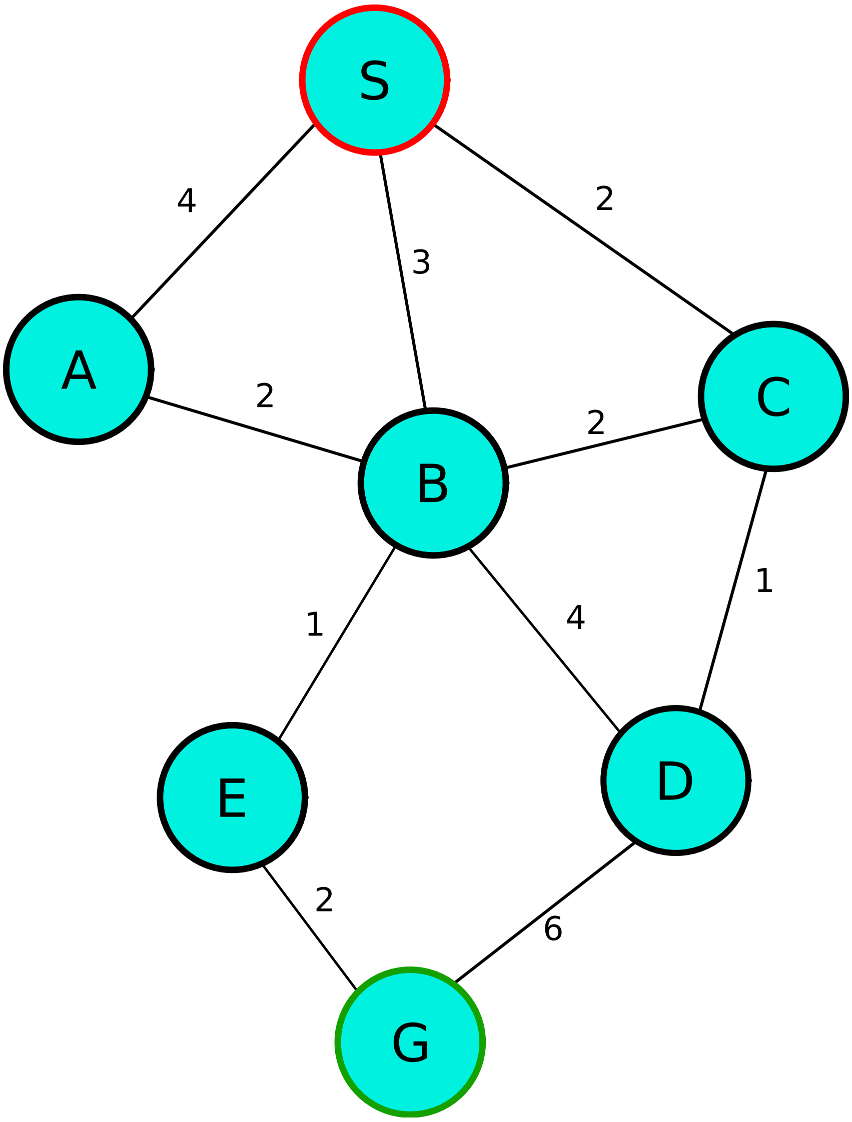


Figure 4. Figure fir Breadth-first search

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.

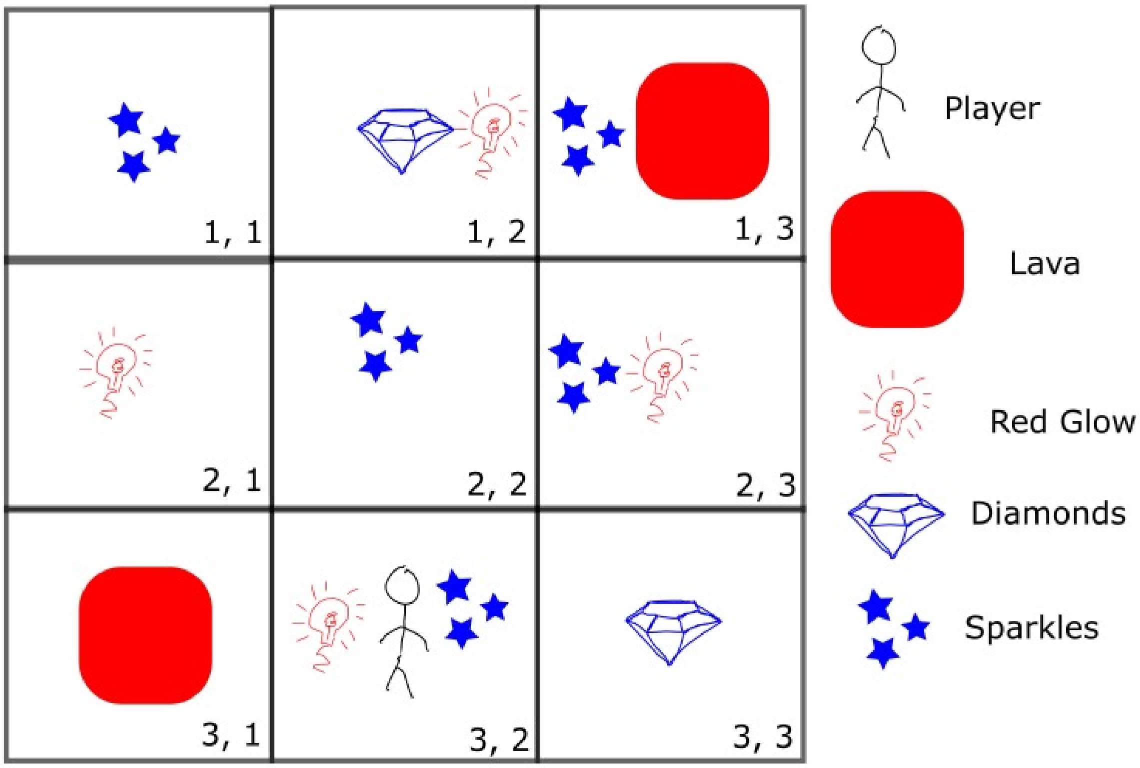


Figure 5. Figure for Logic quest (first task)

1) In this case it doesn't really matter which representation one picks, both can represent the board with the same amount of symbols. Because it doesn't matter I just picked propositional logic.

$Sparkles_{x,y} = ¿$ Square (X, Y) has sparkles
$Diamonds_{x,y}$ = Square X Y has diamonds
$RedGlow_{x,y}$ = Square X Y has a red glow
… and so on for Lava and Player
Board state is then:
$Sparkles_{1,1} \wedge Diamonds_{1,2} \wedge RedGlow_{1,2} \wedge Sparkes_{1,3} \wedge Lava_{1,3} \wedge RedGlow_{2,1} \wedge Sparkles_{2,2} \wedge Sparkles_{2,3}$

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.

b)

1)

$\forall x \forall y \left(Square(y) \land Square(x) \land Adjacent(x, y) \land Has(y, Diamonds) \Rightarrow Sparkles(x) \right)$

c) 2) $\forall x \forall y \left(Square(y) \land Square(x) \land Adjacent(x, y) \land Has(y, Lava) \Rightarrow RedGlow(x) \right)$

d) 3) $\forall x \left(Square(x) \land Has(x, Player) \land Has(x, Lava) \Rightarrow GameOver \right)$

e) 4) $\neg \exists y \left(Diamond(y) \land \neg Has(Player, y) \right) \Rightarrow GameWon$

f) 5)

$\forall x \forall z \left(Square(x) \land Player \land Diamond(z) \land Has(x, Player) \land Has(x, z) \Rightarrow Has(Player, z) \right)$

g) 6)

$\forall x \left(\left(Square(x) \land CanMove(Player, x) \land \exists y \left(Square(y) \land Adjacent(y, x) \land Has(y, RedGlow) \right) \right) \Rightarrow \exists y$

7) $\forall x \forall y \left(Square(x) \land Square(y) \land CanMove(Player, x) \land Has(x, Player) \Rightarrow Adjacent(x, y) \right)$

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.

No, the player can not win. This is because of rule 6. The player can never move to (2, 1) because it has a red glow but does not sparkle, which means that the player can never move to (2, -1)

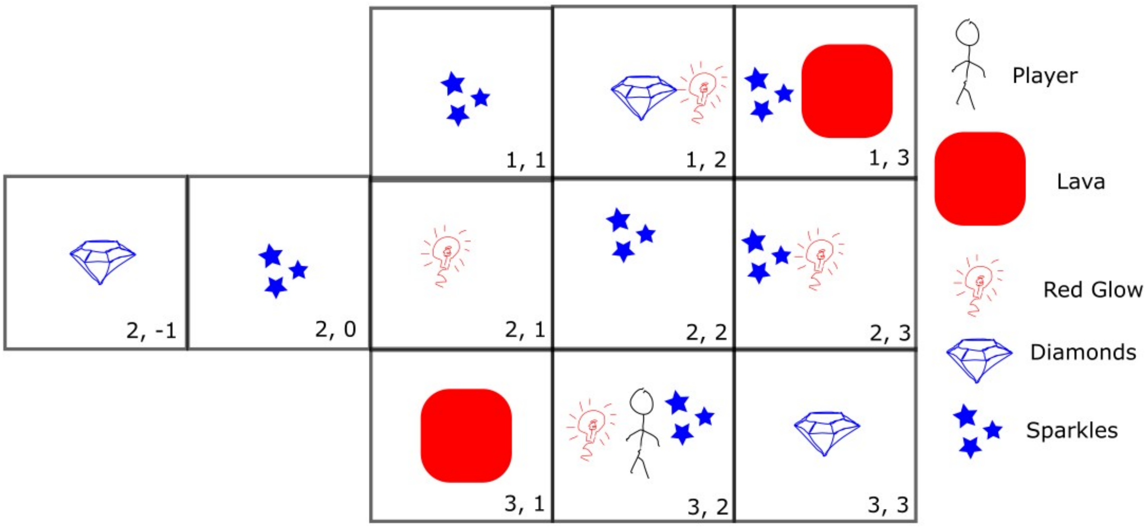


Figure 6. Figure for Logic Quest, Task c.

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

d. $Close(Diamonds, Lava)$

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

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

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

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

4)

a, b, d already in CNF

c: $\forall x \forall y (Person(x) \wedge Afraid(x, y) \Rightarrow \neg Close(x, y))$

gives $\forall x \forall y (\neg Person(x) \vee \neg Afraid(x, y) \vee \neg Close(x, y))$ because implication is equivalent to this

gives $\neg Person(x_1) \vee \neg Afraid(x_1, y_1) \vee \neg Close(x_1, y_1)$ by removing universal quantifiers.

E: Same steps as c gives
$\neg Close(x_2, y_2) \lor \neg Close(y_2, z_2) \lor Close(x_2, z_2)¿$
F: Same steps as c gives
$\neg Close(x_3, y_3) \lor Close(x_3, y_3)$
G: Same steps as c gives
$\neg CanMine(x_4, Diamonds) \lor Close(x_4, Diamonds)$
"Richard does not mine diamonds" is
$\neg CanMine(Richard, Diamonds)$
So refutation is:
H: $CanMine(Richard, Diamonds)$
H and G gives I: Close(Richard, Diamonds)
B and D gives J: $\neg Afraid(Richard, y_1) \lor \neg Close(Richard, y_1)$
A and J gives K: $\neg Close(Richard, Lava)$
K and E gives L: $\neg Close(Richard, y_2) \land \neg Close(y_2, Lava)$
L and I gives M: $\neg Close(Diamonds, Lava)$
M and D gives empty, concluding proof by resolution refutation.

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. $RG \land \neg RG$
   c. $RG \lor \neg RG$
   d. $RG \Rightarrow RG$
   e. $RG \Longleftrightarrow D$
   f. $\neg(RG \Longleftrightarrow D)$
   g. $((RG \lor D) \land \neg(RG \land D)) \Rightarrow RG$

5)
Here RedGlow = RG, Diamonds = D
a. Valid, RedGlow may be false
b. Neither, RedGlow can't be false and true at the same time
c. Valid, RedGlow is always either true or false
d. Valid, equivalent to c
e. Satisfiable, true when RedGlow and Diamonds have same truth value
f. Satisfiable, true when RedGlow and Diamonds do not have same truth value
g. Satisfiable, equivalent to
$(\neg RedGlow \land \neg Diamonds) \lor (RedGlow \land Diamonds) \lor RedGlow$ which is true f.ex. when RedGlow is true and false when RedGlow is false and Diamonds is true.

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).
    a. Domains based on the favorite colours and plant types of persons :
    b. Peter: {LG-Calat1, LG-Calat2, LG-Phil1}
    c. Anette: { V-Calat3, B-Phil2, DG-Calat3 }
    d. Rudolf: {V-Phil1, V-Calat1, B-Phil2 }
    e. Daisy: { V-Calat1, B-Phil2}
    i. Femke: { V-Phil1, V-Calat1, V-Calat2, V-Calat3, DG-Phil1, DG-Calat1, DG-Calat2, DG-Calat3, Y-Phil1, Y-calat1, Y-Calat2, Y-Calat3}
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.

    1. Search tree for “backtracking with forward checking”.

Assign Peter: LG-Calat1, do inference:
Some domains change because Peter now has “reserved” LG colour and Calat1 plan.t Nobody else can take them. Below are all the domains:

A={V-Calat3, B-Phil2, DG-Calat3} ,
R={V-Phil1, B-Phil2}
D has a constraint with P. Plant(P)= Plant(D). Then
D={-) Empty. Backtrack.

Assign Per: **LG-Calat2**
Anette: { V-Calat3, B-Phil2, DG-Calat3 }
Rudolf: {V-Phil1, V-Calat1, B-Phil2 }
Daisy: { V-Calat1}
Femke: { V-Phil1, V-Calat1, ,V-Calat3, DG-Phil1, DG-Calat1, DG-Calat3, Y-Phil1, Y-calat1, Y-Calat3}

Assign Anette= V-Calat3. Do forward check
Rudolf: {B-Phil2 }
Daisy: { } Empty. Stop backtrack

Assign Anette= B-Phil2
Rudolf: {V-Callat1, V-Phil1 }
Daisy: {V-Calat1}

Femke: {V-Phil1, DG-Phil1, Y-Phil1}

Assign Rudolf= V-Calat1
Daisy={} Empty . Backtract

Assign Rudolf= V-Phil1
Daisy={} Empty . Backtract

Assign Anette= DG-Calat3
Rudolf: {V-Phil1, V-Calat1, B-Phil2 }
Daisy: { V-Calat1 }
Femke: { V-Calat1 }. Must have same plant category as Anette, i.e., Calat.

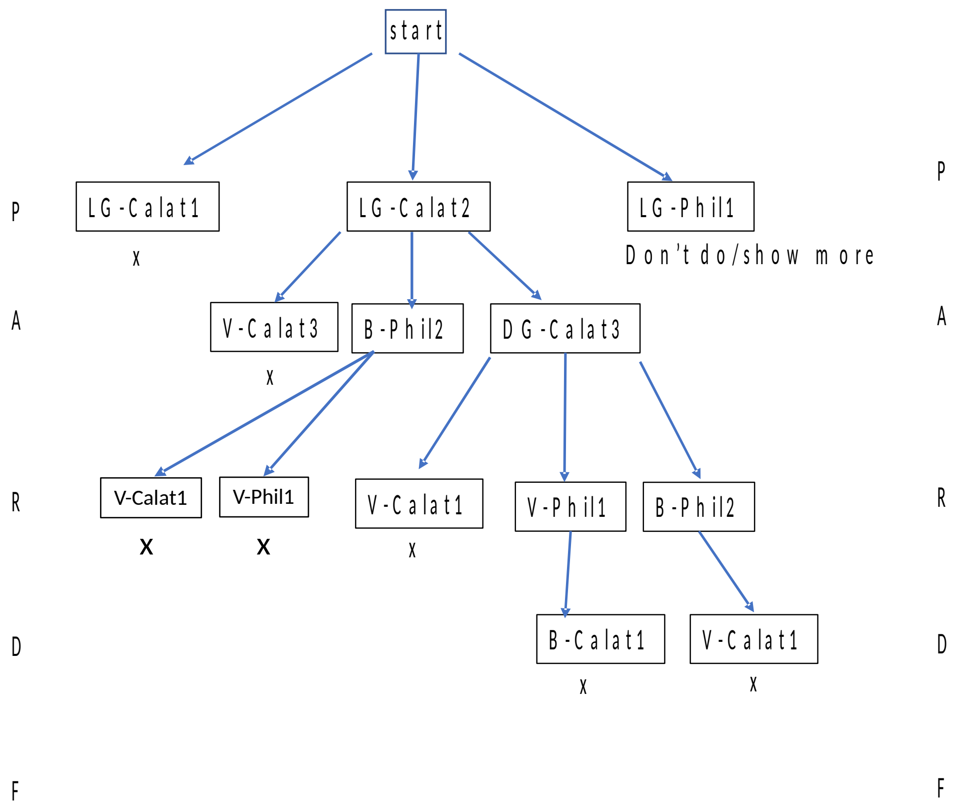
Assign Rudolf= V-Calat1
Daisy: {- }Empty, backtrack

Assign Rudolf= V-Phil1
Daisy: { B-Calat1} must have same same plant category as P, which is Calat
Femke: { Y-Calat1} . must have same plant category as Anette

Assign Daisy= B-Calat1
Femke: { -} Empty, Backtrack

Assign Rudolf= B-Phil2
D: {V-Calat1}
Femke: {Y-calat1, Y-Calat1} same category as Anette.

Assign D= V-Calat1
Femke: {-} Empty backtrack



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.

Assign P: LG-Calat1.
Do inference: Some domains change because person P now has "reserved" LG colour and Calat1 plan. Nobody else can take them. Below are all the domains:

A={V-Calat3, B-Phil2, DG-Calat3} ,
R={V-Phil1, B-Phil2}
D has a constraint with P. It cannot have Calat1 but must have a Calat plant because Plant-category(P) = Plant-category(D). Then
D={ -}. Empty domain. Backtrack.

Assign P= LG-Calat2
Anette: { V-Calat3, B-Phil2, DG-Calat3 }
Rudolf: {V-Phil1, V-Calat1, B-Phil2 }
Daisy: { **V-Calat1**}. Singleton domain, propagate - Constraint on plant category with P.

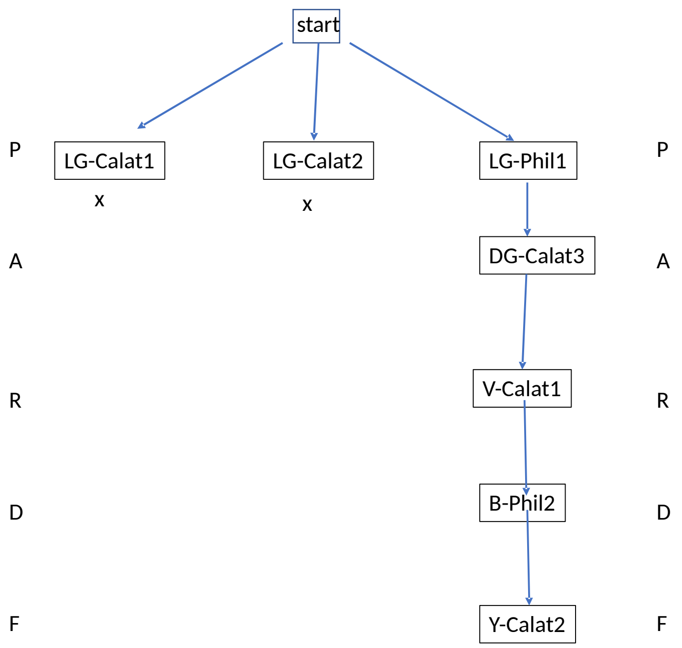
Anette: { B-Phil2, DG-Calat3 }

Rudolf: {**B-Phil2** } Singleton domain, propagate

Anette: {**DG-Calat3**}. Singleton propagate
Femke cannot have LG, V, B, DG. Cannot have Calat2,Calat1,Calat3, and Phil2.
Must have the same plant category as Anette which is Calat.
Femke:{Empty} Backtrack

Assign **P= LG-Phil1**
Anette: {V-Calat3, B-Phil2, DG-Calat3 }
Rudolf: {V-Calat1, B-Phil2}
Daisy: **{ B-Phil2**}-since must have plant in same category as Peter

Anette: { V-Calat3, DG-Calat3 }
Rudolf: {**V-Calat1}** singleton
Anette: {**DG-Calat3** }. Cannot have V because of Rudolf propagation.
Femke: {Y-Calat2}



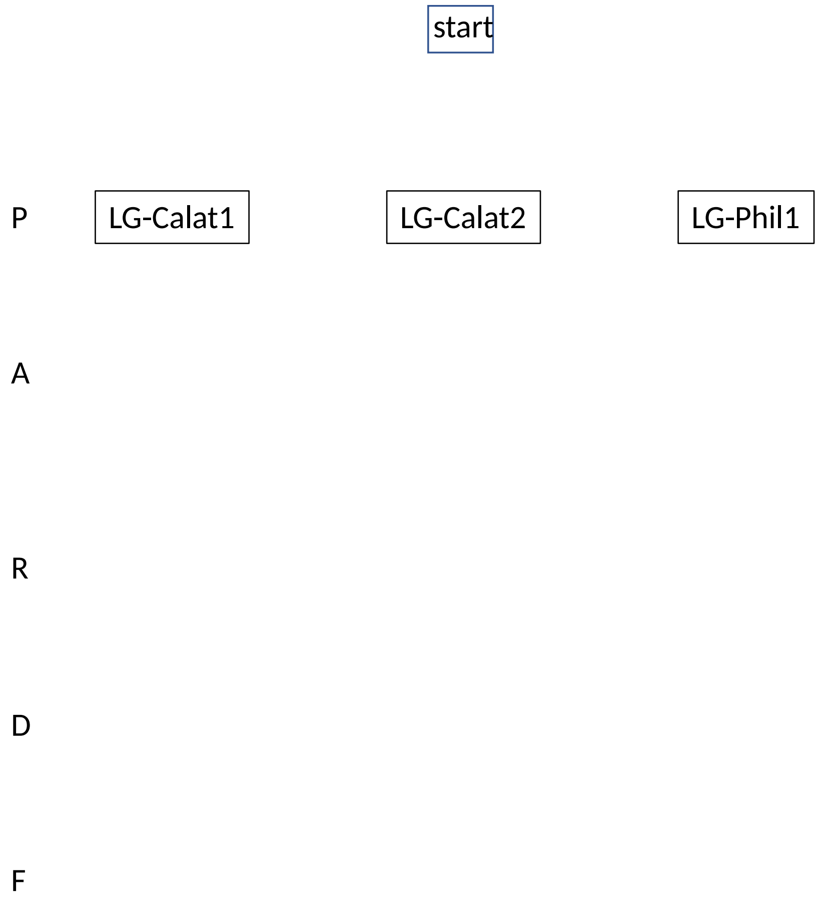


Figure 7. CSP problem. Compete this search tree

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.

a

a. Flaw1. Precondition Smooth(B) of paint(B) is not satisfied
b. Flaw2. putOn(A,B) threatens paint(B) –because it makes A not free
c. Flaw3. putOn(A,B) threatens paint(A) - –because it makes B not free
d. Flaw4. putOn(A,B) threatens sand(A) - because it makes A not free
e. So, need for ordering links.

b

Flaw 1: add action. Sand(B)

Flaw 2: paint B must be executed before putOn(A,B), i.e., an ordering link is added

Flaw 3: paint B must be executed before putOn(A,B), i.e., an ordering link is added

Flaw 4: putOn(A,B) comes after sand(A) as well as newly added sand(B)

c

(sand(A) OR sand(B)) ; (Paint(A) OR paint(B)) ; putOn(A,B)

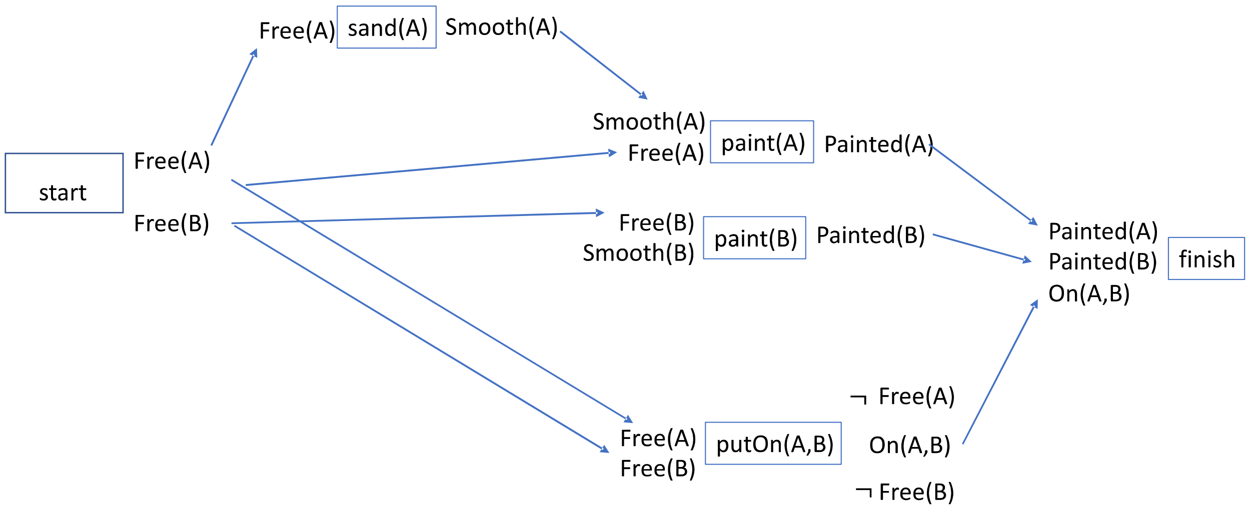


Figure 8. Figure for planning

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?
   a. No constraints, they can be executed in parallel.

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.

Both return TIK as the plan. Its postconditions (C and D) satisfies the goal $C \wedge D$.

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.
   a. The Nash equilibrium is when all students chose the same integer. No student will have an incentive to move away from that number because moving to any other number will end up in zero payoff. There is no other Nash because any student who has chosen a less-often-chosen number would benefit from switching to the most-often one. Also if there are two numbers that are most often, students who have chosen these will benefit from switching to the other most-often number.

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.
All strategy profiles except (B,B) are pareto optimal. (B,A) is also social optimum.

Agent 1 \ Agent 2	A	B
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:

Agent1 \ Agent2	S	T	R
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.

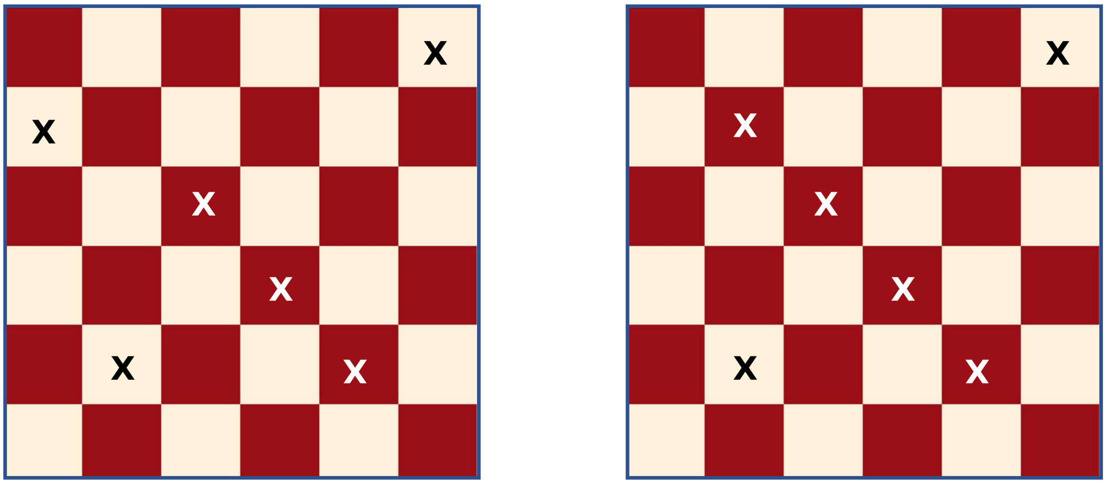
a. We eliminate a and b because c dominates them. Then on the remaining matrix, we can eliminate S and R as they both are dominated by T. Then the solution is (c,T).

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.
      a. Eval(a)= 1- # attacking pairs(a)= 1-5= -4
      b. Eval(b) = 1-# attacking pairs(b)= 1-9= -8
   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.
      See textbook p.126, Figure 4.5. Simulated Annealing algorithm. The alg decides to move if Delta-E> 0. If smaller, then moves with probability $e^{(Delta-E/T).}$
      Delta E= (1-9)-(1-5)= -4. Moves with probability 1/e.



(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.

1) **PEAS and Characteristics of the Environment**
Environment: partially observable, deterministic (but may be stochastic if the sensor is not good and the robot may be blown away by the mine), continuous, static, sequential (mines are most probably places with a some distance between them. If one found, then the next one will be at least in 30 cm distance). The answers may differ based on the assumptions. So, various answers are accepted here.

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.

2) **Translation from Semantic Networks to Logic**

Translate the sematic network representations shown in the figure to logic representations. The following is the outset. I accepted also variations such as Isa(elephant,mammal), color(elephant, grey) type of answers without quantification.

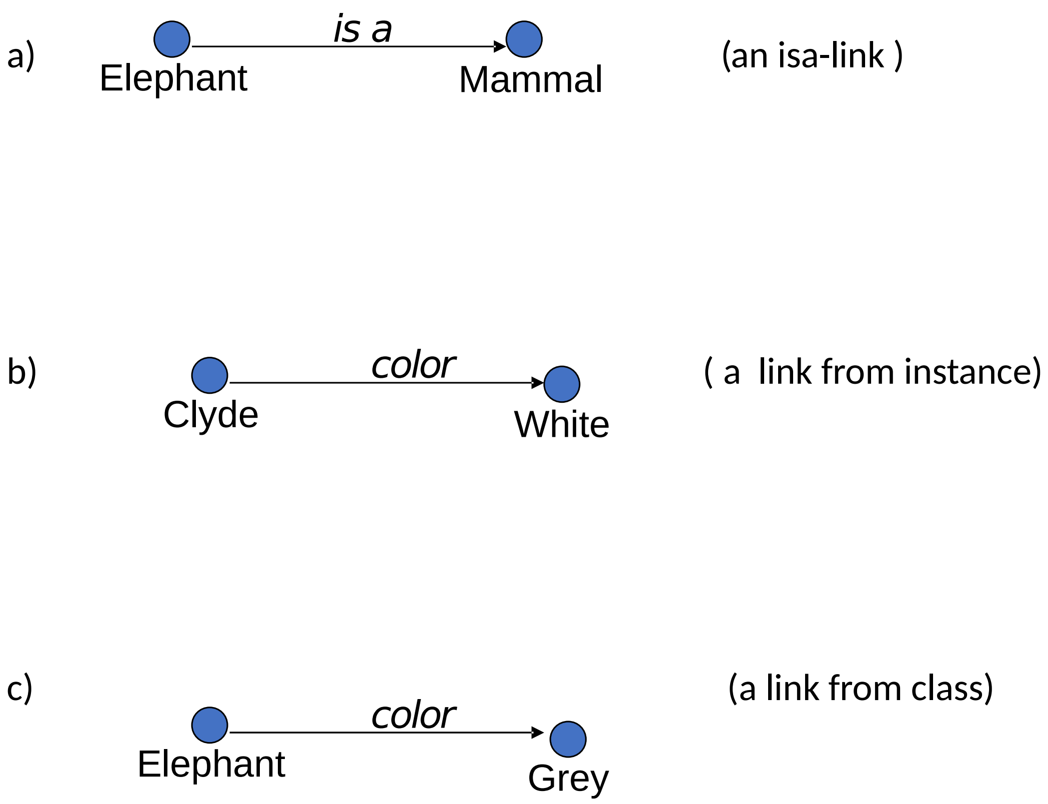


Figure 12. Translation from SN to Logic

KONT EKSAMEN AUGUST 2021--- QUESTIONS

Weighting: There are 5 questions in this exam. The weights are as following: Q1:25 points, Q2:25 pts, Q3: 25pts, Q4: 10pts, and and Q5 15 pts each

PROBLEM 1 - LOGIC

Given the following three sentences, you will prove ¬ S is true.

$P \wedge Q$ (1)
$P \rightarrow \neg(Q \wedge R)$ (2)
$S \rightarrow R$ (3)

a) Show $\neg S$ is true by using the inference and the re-writing rules.

b) Show that $\neg S$ is true by enumeration method on a Truth Table.

c) Show that $\neg S$ *is true by resolution refutation.*

d) Use resolution refutation to prove Green(Sophie) given the information below. You must first convert each(all) sentence into CNF. Clearly show Skolemizations. Show only the applications of the resolution rule that lead to the desired conclusion. For each application of the resolution rule, show the unification bindings, θ.

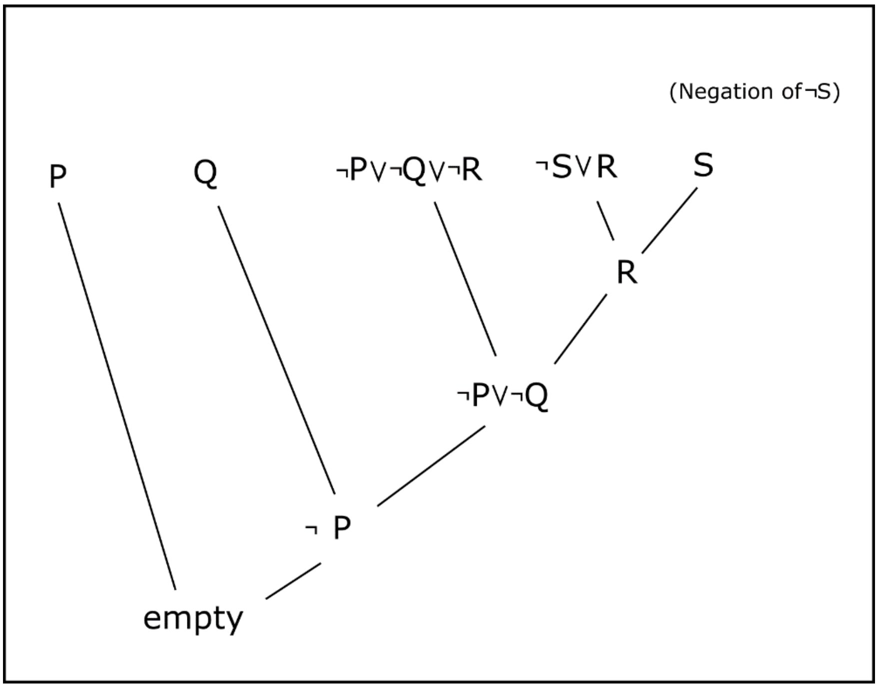
- Electric(Tesla)
- Drives(Sophie, Tesla)
- ∀x Green(x) ↔ Bikes(x) ∨ [∃y : Drives(x, y) ∧ Electric(y)]

a)

1. $P \wedge Q$ given (premise)
2. P (from 1, decomposing a conjunction)
3. Q (from 1)
4. $P \rightarrow \neg(Q \wedge R)$ given
5. $\neg(Q \wedge R)$ (from 2,4)
6. ¬Q ∨ $\neg R$ (from 5)
7. $\neg R$ (from 3,6)
8. $S \rightarrow R$ given
9. $\neg S$ (from 7,8)

b) Draw the truth table and see there is one row where 1,2, and 3 is true and $\neg S$ *is also true there.*

c)



d) First, we need to convert the definition of Green into CNF.

a) ∀x : Green(x) ⇔ Bikes(x) ∨ [∃y : Drives(x, y) ∧ Electric(y)]

Break the double-implication into 2 conjoined implications

b) ∀x : [Green(x) → Bikes(x) ∨ [∃y : Drives(x, y) ∧ Electric(y)]] ∧

[[Bikes(x) ∨ [∃y : Drives(x, y) ∧ Electric(y)]] → Green(x)]

Convert implications to disjunctions

c) ∀x : [¬Green(x) ∨ Bikes(x) ∨ [∃y Drives(x, y) ∧ Electric(y)]] ∧

¬[Bikes(x) ∨ [∃y Drives(x, y) ∧ Electric(y)] ∨ Green(x)

Move negations inward

d) ∀x : [¬Green(x) ∨ Bikes(x) ∨ [∃y : Drives(x, y) ∧ Electric(y)]] ∧

¬Bikes(x) ∧ ¬[∃y Drives(x, y) ∧ Electric(y)] ∨ Green(x)

Continue moving negations inward

a) ∀x : [¬Green(x) ∨ Bikes(x) ∨ [∃y Drives(x, y) ∧ Electric(y)]] ∧

¬Bikes(x) ∧ [∀y ¬Drives(x, y) ∨ ¬Electric(y)] ∨ Green(x)

Skolemizing produces an F(x) in place of the existential-quantified y:

b) ∀x : [¬Green(x) ∨ Bikes(x) ∨ [Drives(x, F(x)) ∧ Electric(F(x))]] ∧
¬Bikes(x) ∧ [∀y : ¬Drives(x, y) ∨ ¬Electric(y)] ∨ Green(x)

Remove the universal quantifications, since all remaining variables are universally quantified.

c) [¬Green(x) ∨ Bikes(x) ∨ [Drives(x,F(x)) ∧ Electric(F(x))]] ∧

¬Bikes(x) ∧ [¬Drives(x, y) ∨ ¬Electric(y)] ∨ Green(x)

Distribute the disjunction in the first half

d) [¬Green(x) ∨ Bikes(x) ∨ Drives(x,F(x))] ∧
[¬Green(x) ∨ Bikes(x) ∨ Electric(F (x))] ∧
¬Bikes(x) ∧ [¬Drives(x, y) ∨ ¬Electric(y)] ∨ Green(x)

Distribute the disjunction in the second half to produce a conjunction of 4 disjuncts (CNF).

e) [¬Green(x) ∨ Bikes(x) ∨ Drives(x,F(x))] ∧

[¬Green(x) ∨ Bikes(x) ∨ Electric(F (x))] ∧

[Green(x) ∨ ¬Bikes(x)] ∧
[¬Drives(x, y) ∨ ¬Electric(y) ∨ Green(x)]

Next, combine these 4 clauses with the other givens and add in the negation of the goal sentence: Green(Sophie). Then keep applying the resolution rule until θ = False is derived, indicating the contradiction.

a. ¬Green(x) ∨ Bikes(x) ∨ Drives(x,F(x)) Given

b. ¬Green(x) ∨ Bikes(x) ∨ Electric(F(x))] Given

c. Green(x) ∨ ¬Bikes(x)] Given

d. ¬Drives(x, y) ∨ ¬Electric(y) ∨ Green(x) Given

5. Electric(Tesla) Given

6. Drives(Sophie,Tesla) Given
7. ¬Green(Sophie) (Assuming negation of target sentence)
8. ¬Drives(x, Tesla) ∨ Green(x) (Resolving 4 and 5 with θ = {y/Tesla})

9. Green(Sophie) (Resolving 6 and 8 with θ = {x/Sophie})

10. (Resolving 7 and 9 with θ = {})

Notice that only 1 of the 4 clauses derived from the definition of Green was used to prove the target sentence.

PROBLEM 2 --INFORMED AND UNINFORMED SEARCH

You are given the following graph in the figure where S is the start node and there are three goal nodes, G1, G2 and G3. You are asked to use some search algorithms on this graph (algorithms a-d listed below.) You will use tree search versions of these algorithms that avoid re-expanding the nodes that are already expanded. In case of a tie, break will be done in alphabetical order. In all the algorithms, you don't need to continue after finding a goal, i.e, you don't need to search for more than one goal.

For each of the asked algorithms,

- write down the expanded nodes in the order of expansion,
- write down the solution path and its cost.
- Is the found path optimal? Discuss the optimality of the algorithm.

The algorithms are:

a) Uniform cost search
b) Breadth first search
c) Depth first search
d) A* search. The heuristic values are as follows:
   h(S)=5
   h(A)=7
   h(B)=3
   h(C)=4
   h(D)=6
   h(E)=5
   h(F)=6
   h(G1)=h(G2)=h(G3)=0

a) Uniform cost:

Expanded nodes: SADBCE G2

Solution path: S D C G2

Path cost: 13. Optimal path. Uniform cost search is optimal when there are no negative path costs.

b) Breadth first:

Expanded: S A G1. (goal check is when childs are generated)

S. Path: S A G1

Path cost: 14. Not optimal. BFS is cost optimal only when the steps costs are identical

c) Depth first

Expanded nodes: S A B C F D E G3

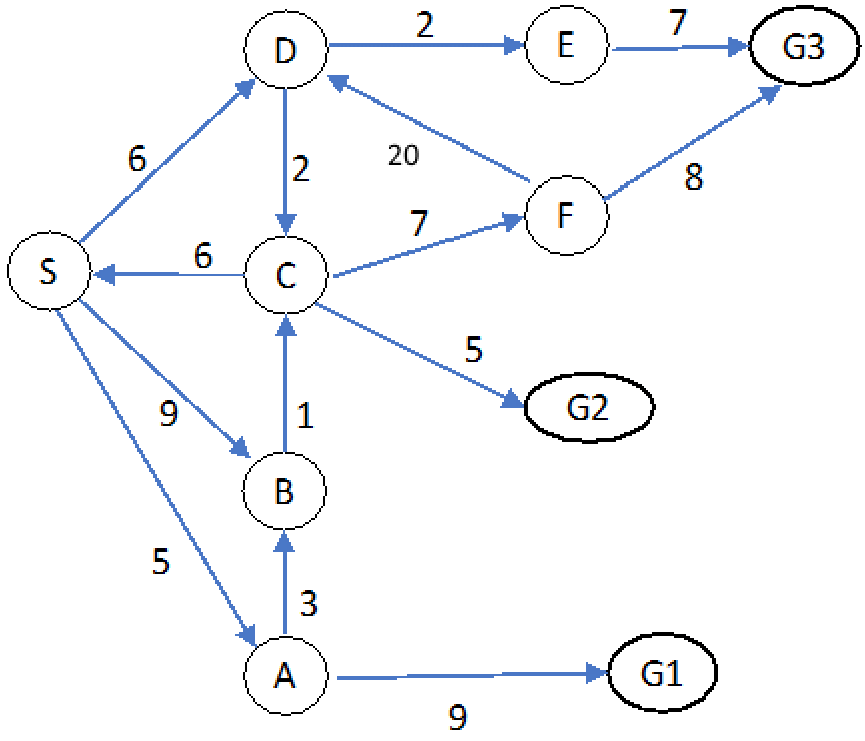
Solution cost: 45

d) A*

Expanded nodes: S A B D C E G2

Solution path: S D C G2

Path cost: 13. Optimal.



PROBLEM 3 ---CSP CROSS WORD PUZZLE

A **crossword puzzle**, is a popular type of puzzle that uses words.

A crossword is made up of a black/blue and white squares, called a grid. Next to the grid is a list of clues. The answer to each clue is a word. The place in the grid where the answer to each clue should go is shown by a number and the direction in which the answer appears, for example "1 Across", or "4 Down".

You are given the 4x5 grid on the top of the page. Unlike traditional crossword puzzles, you are not given clues but a list of words to choose from to fill in the grid.

Here is the list of words: *astar, happy, hello, hoses, live, load, loom, peal, peel, save, talk, anon, nerd, tine, ant, oak, old, ten, run.*

There are 4 areas in the grid/puzzle with consecutive white squares where the words will be placed: 1-accross (i.e., from left to write), 2-down (write from top to downward), 3-down and 4-across. Blue squares are blocked out meaning that characters cannot be placed on them.

"Instructions" on the grid, e.g., 2-down, indicate the position where a word starts and the direction it will continue until it hits a blue square or the edge of the grid.   Note that the number in an instruction does not indicate the length of the word. The length of the word is determined by the number of consecutive white squares in the given direction. For example, the word starting at 1-accross is five character long. A solution is a grid with correctly placed words -and no empty white squares. Each word will be used only once.

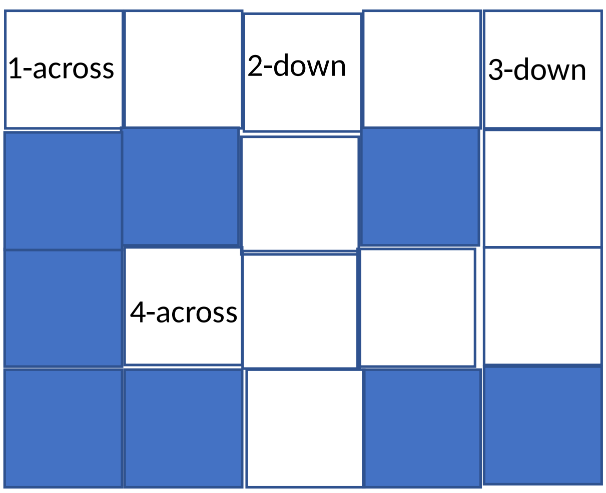
Consider this as a constraint satisfaction problem and:

a) Write down the variables (V1, V2, ..) and describe what they correspond to in the grid shown in the figure. Draw the constraint graph.

b) How many constraints are there in this problem and what are they?

c) For each variable write down the domain satisfying the node consistency

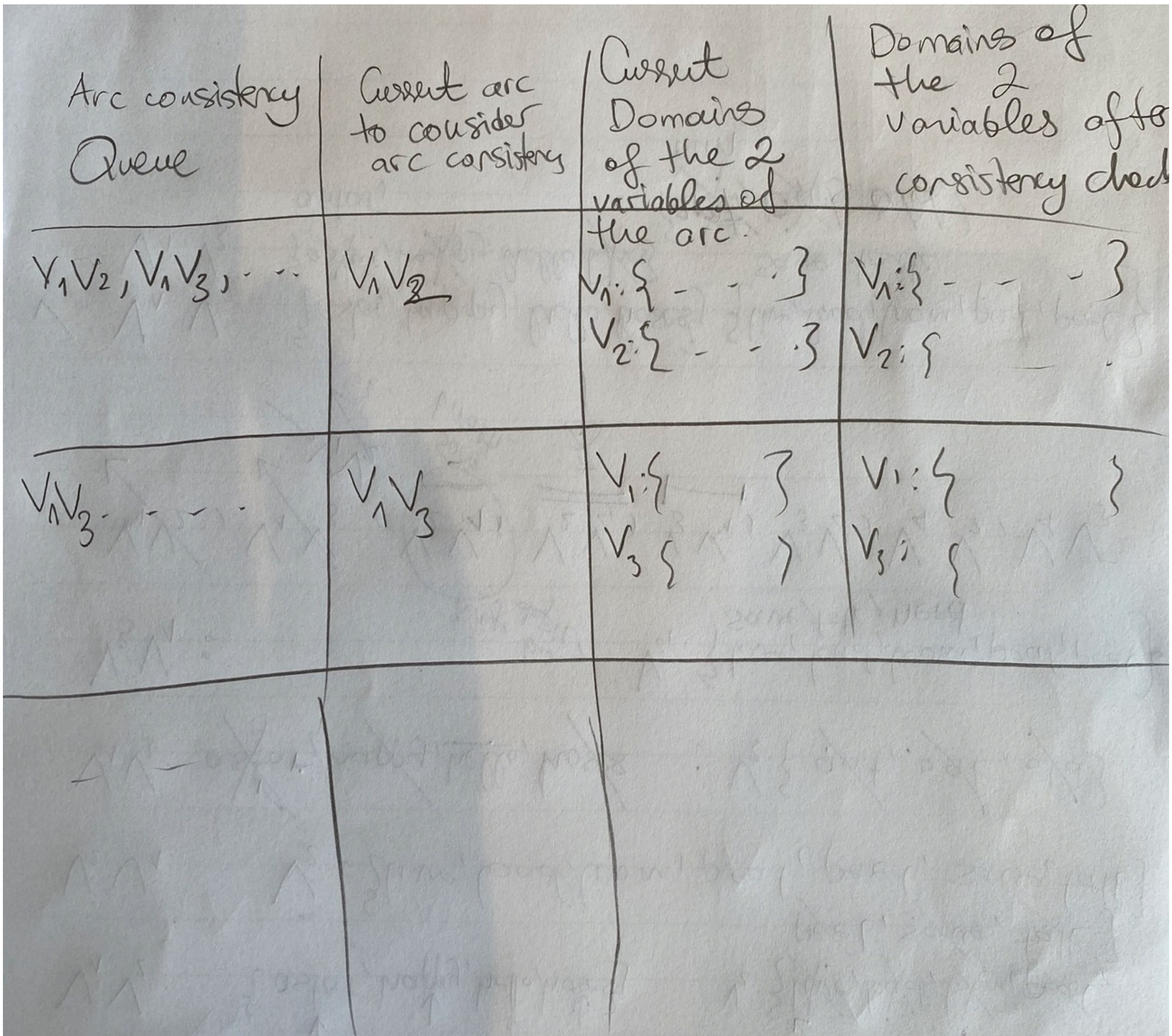
d) Apply arc consistency algorithm AC-3. Write down the domains of the variables after AC. Write down the domains of the variables after AC-3. When applying AC-3, assume a queue (FIFO) of edges of the constraint graph initially sorted in the ascending order, starting from V1, i.e. V1V2, V1V3,.... In order to show your work, fill in the table (similar to the one) in the figure on left.



e) If AC-3 finds a solution, what is it? Provide the filled-out crossword puzzle. If it does not find any solution, explain why

Consider this as a constraint satisfaction problem and:

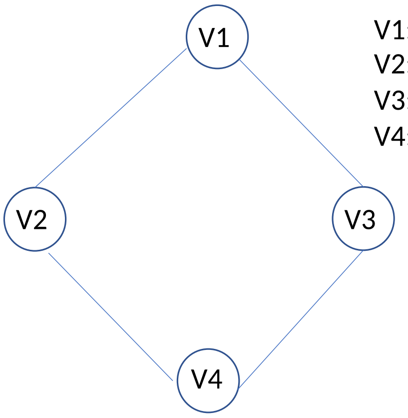
e) Write down the variables (V1, V2,…), what each corresponds to on the grid shown in Figure xxxxxx, and draw the constraint graph.
f) How many constraints are there in this problem and what are they?
g) For each variable write down the domain satisfying the node consistency.
h) Apply arc consistency algorithm AC-3. Write down the domains of the variables after AC-3. When applying AC-3, assume a queue (i.e.,FIFO) of edges of the constraint graph initially sorted in the ascending order, starting from V1, i.e., V1V2, V1V3,… In order to show your work, fill in the table (similar to the one) below.



Arc consisteny Queue	Current considered arc	Domains of arc variables before arc consistency check	Domains of arc variables after arc consistency check
V1V2, V1V3, … (but write out fully)	V1V2	V1:{Write out domain} V2:{…}	V1:{…} V2:{…}
V1V3, …	V1V3,	V1:{…} V3:{…}	V1:{…} V3:{…}

i) If AC-3 finds a solution, what is it (provide the filled-out crossword puzzle, if any)? If it does not find any solution, explain why.

a)



V1: 1-across
V2: 2-down
V3: 3-down
V4: 4-accross

b) C1: V1 has 5 letters
C2: V2 has 3 letters
C3: V3 has 3 letters
C4: V4 has 4 letters
C5: 3rd letter of V1 is the same letter as the first letter of V2
C6: 5th letter of V1 is the same letter as the first letter of V3
C7: 2nd letter of V4 is the same letter as 3rd letter of V2
….
….

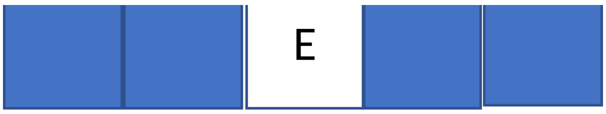
c) Domains, according to node consistency:

V1 ----Domain1={ astar, happy, hello, hoses}
V2 ----Domain2={ live, load, loam, peal, peel, save, talk, anon, nerd, tine }
V3 ----Domain3={ ant, oak, old, run, ten}
V2 ----Domain2={ live, load, loam, peal, peel, save, talk, anon, nerd, tine}

d)

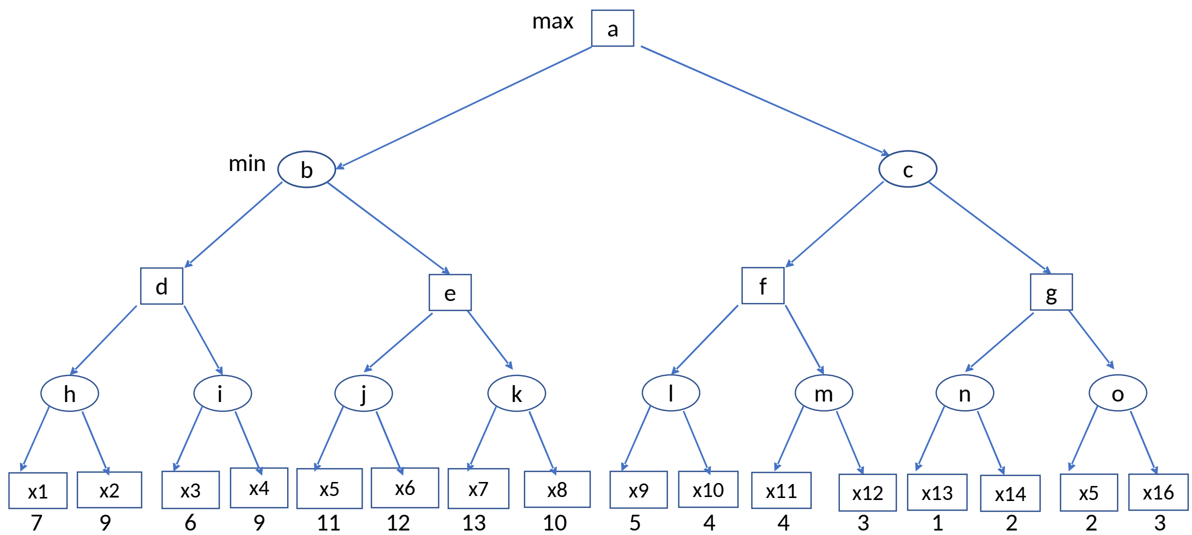
Arc consistency Queue	**Current considered arc**	**Domains of arc variables before arc consistency check**	**Domains of arc variables after arc consistency check**
V1V2, V1V3, V2V1, V2V4, V3V1, V3V4, V4V2, V4V3	V1V2	V1:{astar, happy, hello, hoses} V2:{live, load, loom, peal, peel, save, talk, anon, nerd, tine}	V1:{astar, happy, hello, hoses} V2:{live, load, loom, peal, peel, save, talk, anon, nerd, tine}
V1V3, V2V1, V2V4, V3V1, V3V4, V4V2, V4V3	V1V3	V1:{astar, happy, hello, hoses} V3:{ant, oak, old, ten, run}	V1:{astar, hello} V3:{ant, oak, old, ten, run}
V2V1, V2V4, V3V1, V3V4, V4V2, V4V3	V2V1	V2:{live, load, loom, peal, peel, save, talk, anon, nerd, tine} V1:{astar, hello}	V2:{live, load, loom, talk, tine} V1:{astar, hello}
V2V4, V3V1, V3V4, V4V2, V4V3, V1V2	V2V4	V2:{live, load, loom, talk, tine} V4:{live, load, loom, peal, peel, save, talk, anon, nerd, tine}	V2:{load, loom, tine} V4:{live, load, loom, peal, peel, save, talk, anon, nerd, tine}
V3V1, V3V4, V4V2, V4V3, V1V2	V3V1	V3:{ant, oak, old, ten, run} V1:{astar, hello}	V3:{oak, old, run} V1:{astar, hello}
V3V4, V4V2, V4V3	V3V4	V3:{oak, old, run}	V3:{oak, old, run}

V1V2, V1V3		V4:{live, load, loom, peal, peel, save, talk, anon, nerd, tine}	V4:{live, load, loom, peal, peel, save, talk, anon, nerd, tine}
V4V2, V4V3, V1V2 V1V3, V2V4	V4V2	V4:{live, load, loom, peal, peel, save, talk, anon, nerd, tine} V2:{load, loom, tine}	V4:{load, loom, save, talk, anon} V2:{load, loom, tine}
V4V3, V1V2, V1V3, V2V4, V3V4	V4V3	V4:{load, loom, save, talk, anon} V3:{oak, old, run}	V4:{load, talk, anon} V3:{oak, old, run}
V1V2, V1V3, V2V4, V3V4	V1V2	V1:{astar, hello} V2:{load, loom, tine}	V1:{astar, hello} V2:{load, loom, tine}
V1V3, V2V4, V3V4	V1V3	V1:{astar, hello} V3:{oak, old, run}	V1:{astar, hello} V3:{oak, old, run}
V2V4, V3V4	V2V4	V2:{load, loom, tine} V4:{load, talk, anon}	V2:{load, loom, tine} V4:{load, talk, anon}
V3V4	V3V4	V3:{oak, old, run} V4:{load, talk, anon}	V3:{oak, old, run} V4:{load, talk, anon}



PROBLEM 4 ---- ADVERSARIAL SEARCH

The following figure shows a game tree.

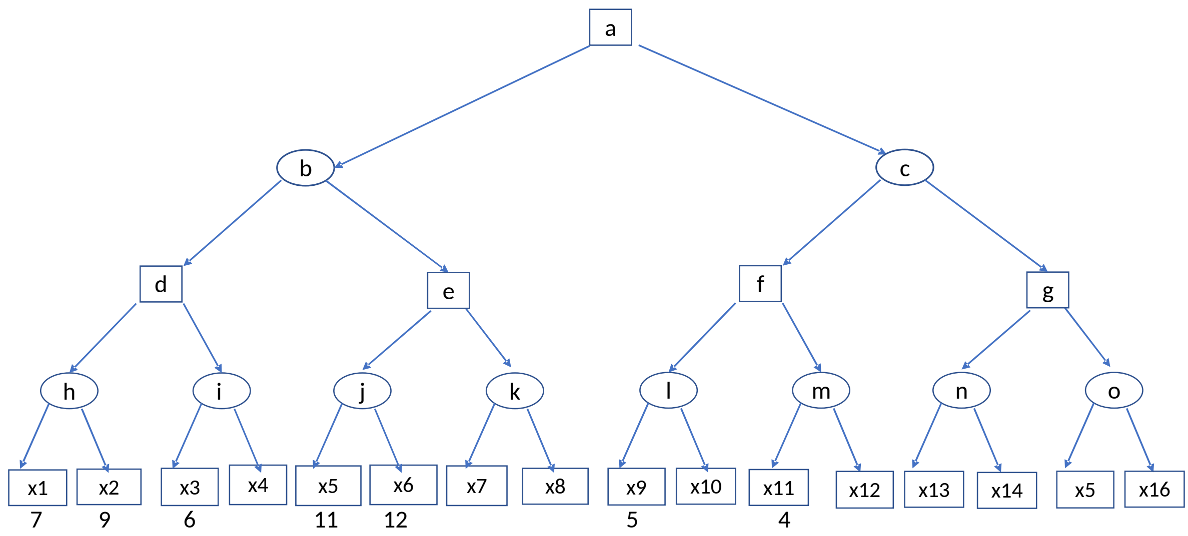


f) Apply Minimax algorithm and find the values of all the nodes above the leaf nodes. What is the value of the solution for the agent *max*?

g) Apply alpha-beta pruning algorithm to find which nodes can e pruned. Write down the names of the nodes (the letters in the square and ellipses), either leaves or the nodes of which all children are pruned. For example if 11 and 12 are pruned then it is sufficient to write node *j* only.

a) H=7,i<=6, d=7, j=11, e>=11, b= 7, c<=5, f<=5, l <=5, m<=4. solution=7

b) x4, k, x10, x12, and g are pruned



PROBLEM 5--- GAME THEORY

In an auction two agents are competing to obtain a porcelain vase. This is a type of auction where **all** bidders simultaneously submit **sealed bids** to the auctioneer so that **no** bidder knows how much the **other** auction participants have **bid.**

The allowed bids are \$0, \$10, \$20, \$30, \$40, \$50.
The porcelain is worth \$40 to the agent A1 and \$30 to the agent A2. The highest bidder wins the porcelain. In case of a tie, A1 gets the vase.
According to the rules of this auction, the winner pays a price *p* which is whatever the other agent bids. So, if the value of the porcelain for agent i is *x* and agent *i* wins the vase her payoff is x-p. If she does not win the vase her payoff is zero.
Answer the following questions:
a. Write down the payoffs of the agents as a matrix – illustrating agents, actions(bids), and the payoffs
b. Is there a strictly dominant strategy equilibrium of this game? Explain.
c. Is there a weakly dominant strategy equilibrium of this game? Explain.
d. What are the action profiles that survive Iterated Elimination of Strictly Dominated Actions? Explain how do the eliminations take place.
e. What are the action profiles that survive Iterated Elimination of Weakly Dominated Actions? Explain how do the eliminations take place.
f. What is the solution if the game is solvable using the 'dominance' concept. (i.e., dominance solvable). Explain your answer.

a) N={A1, A2}, Domains of A1=A2 ={0,10,20,30,40,50}, and the payoff fns are are specified by the following matrix

A1, Agent2	0	10	20	30	40	50
0	40, 0	0, 30	0, 30	0, 30	0, 30	0, 30

10	40, 0	30, 0	0,20	0, 20	0, 20	0, 20
20	40, 0	30, 0	20, 0	0, 10	0, 10	0, 10
30	40, 0	30, 0	20, 0	10, 0	0, 0	0, 0
40	40, 0	30, 0	20, 0	10, 0	0, 0	0, -10
50	40, 0	30, 0	20, 0	10, 0	0, 0	-10 , 0

b) There is no weakly dominant strategy eq. as neither player has a weakly dominant action. Notice that for both players, actions 30 and 40 weakly dominate every other action. But not wach other.

c) D) There is no strictly dominated action for either player and and hence all the action profiles survive IESD actions

d) We can eliminate the weakly dominated actions in the following order:

A:0

A2:0

A1: 50

A2: 50

A1:10

A2: 10

A1: 20

Which leads to the following set of outcomes {30,40} x {20,30,40}. However, tehre are other orders of elimination which lead to different outcomes.

e) The game is not dominance solvable.

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.

   ANSWER: False. It takes the first path to goal it finds.

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.

   ANSWER: True

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.
   Answer= false. It is b^d
4. True or False: $(A \Longleftrightarrow B) \vDash (\neg A \vee B)$

   ANSWER: True

5. Linear planning is incomplete.
   Answer: True, Linear planning is incomplete. Think for example about Sussman's
   anomaly.

6. Simple "hill climbing" algorithm is perfect to solve constraint satisfaction problems.
   ANSWER: False. It can get stuck at local minima and fail to find a solution.

7. A sound logical reasoning process is not necessary in order to pass the Turing test.
   ANSWER: True. Humans don't always use sound logical reasoning

8. Depth-first tree search algorithm always expands at least as many nodes as an A* tree search algorithm with admissible heuristic does.

   ANSWER: False: a lucky DFS might expand exactly d nodes to reach the goal. A* largely dominates any graph-search algorithm that is guaranteed to find optimal solutions.

9. The set consisting of "mammal" and "non-mammal" categories is both a disjunctive and an exhaustive decomposition of the category "animal".

Answer: True.

10. Explainability is an ethical problem in AI which domain knowledge may help to solve. Answer: True

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. $\forall x\ [\exists y\ (Animal(y) \wedge Kills(x,y)] \Rightarrow \neg(\exists z\ Loves(z,x))$
3. $\forall x\ [Animal(x) \Rightarrow Loves(Sofie, x)]$
4. $Kills(Sofie,Kismet) \vee Kills(CarAccident,Kismet)$
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.

ANSWER TO PROBLEM – FOL

C
N
F:

Everyone who loves all animals is loved by someone.

In
FOL

$\forall x\{[\forall y[Animal(y) \Rightarrow Loves(x,y)]] \Rightarrow [\exists z\ Loves(z,x)]\}$

Remove Implications
∀x {[¬[∀y {Animal(y) => Loves(x,y)}]] ∨ [∃z Loves(z,x)]}
∀x {[¬[∀y {¬ Animal(y) ∨ Loves(x,y)}]] ∨ [∃z Loves(z,x)]}
Move negation inward
∀x {[∃y {¬ ¬ Animal(y) ∧ ¬ Loves(x,y)}] ∨ [∃z Loves(z,x)]}
∀x {[∃y Animal(y) ∧ ¬ Loves(x,y)] ∨ [∃z Loves(z,x)]}

Skolemize
∀x{[Animal(F(x))∧¬Loves(x,F(x))]∨ [Loves(G(x),x)]}

Drop universal quantifier
[Animal(F(x)) ∧ ¬ Loves(x,F(x))] ∨ [Loves(G(x),x)]
Use distributive law (and get two clauses)
(Animal(F(x)) ∨ Loves(G(x),x)) AND (¬ Loves(x,F(x)) ∨ Loves(G(x),x))

Anyone who kills an animal is loved by no one. Transfer to FOL
∀x {[∃y (Animal(y) ∧ Kills(x,y)] =>¬(∃z Loves(z,x))

Remove Implications
∀x {¬[∃y (Animal(y) ∧ Kills(x,y)] ∨ ¬(∃zLoves(z,x))}

Move negations inwards
∀x {[∀ y ¬Animal(y) ∨ ¬ Kills(x,y)] ∨ (∀z ¬ Loves(z,x))}

Remove quantifiers
¬Animal(y) ∨ ¬ Kills(x,y) ∨ ¬ Loves(z,x)

Sofie loves all animals.
FOL form
∀x [Animal(x) => Loves(Sofie, x)]

Remove implications
∀x [¬Animal(x) ∨ Loves(Sofie, x)]

Remove quantifier
¬Animal(x) ∨ Loves(Sofie, x)

Either Sofie or CarAccident killed the cat, who is named Kismet.
FOL form
Kills(Sofie,Kismet) ∨Kills(CarAccident,Kismet), Cat(Kismet)

Bakground knowledge: All cats are animals: ∀x Cat(x) => Animal(x)

RESOLUTION:

Cat(Kismet) , ¬Cat(x) ∨ Animal(x)

Unify(Cat(Kismet), ¬Cat(x)) = { x/Kismet }

First line thus resolves to:

Animal(Kismet)

Kills(Sofie,Kismet) ∨ Kills(CarAccident,Kismet), ¬Kills(CarAccident,Kismet)
Resolves to:

Kills(Sofie,Kismet)

¬Animal(y) ∨ ¬ Kills(x,y) ∨ ¬ Loves(z,x), Animal(Kismet)
Unify(Animal(Kismet), ¬Animal(y)) = {y/Kismet}

Resolves to:

¬ Kills(x,Kismet) ∨ ¬ Loves(z,x),

¬ Loves(x,F(x)) ∨ Loves(G(x),x), ¬Animal(z) ∨ Loves(Sofie, z)

Unify(¬ Loves(x,F(x)) , Loves(Sofie, z)) = { x / Sofie, z / F(x)}

Resolvent clause is obtained by substituting the unification rule

Loves(G(Sofie),Sofie) ∨ ¬Animal(F(Sofie))

Animal(F(x)) ∨ Loves(G(x),x), Loves(G(Sofie),Sofie) ∨ ¬Animal(F(Sofie))

Unify(Animal(F(x)) , ¬Animal(F(Sofie)))= { x / Sofie}

Resolvent clause is obtained by substituting the unification rule

Loves(G(Sofie),Sofie)

¬ Kills(x,Kismet) ∨ ¬ Loves(z,x), Loves(G(Sofie),Sofie)

Unify(¬ Loves(z,x), Loves(G(Sofie),Sofie)) = { x / Sofie, z / G(Sofie)}

Resolvent clause is obtained by substituting the unification rule

¬ Loves(G(Sofie),Sofie)

¬ Loves(G(Sofie),Sofie), Loves(G(Sofie),Sofie)

Resolvent clause is empty. Proof succeeded

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.

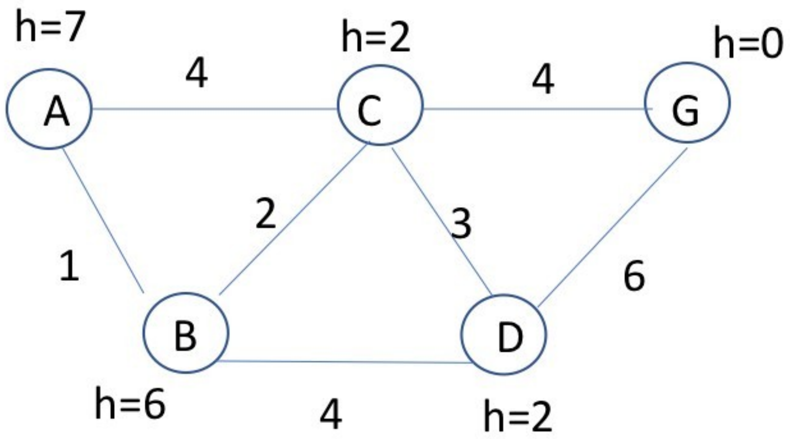


Figure 1 Graph for the question. A is the start node, and G is the goal node.

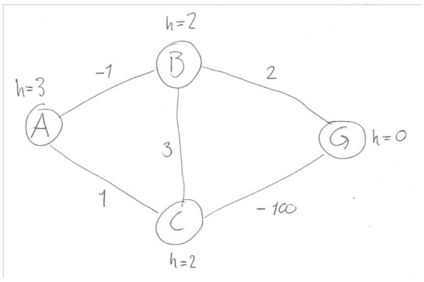
```
WHILE FRONTIER is not empty
    N = FRONTIER.popLowestF()
    IF state of N= GOAL RETURN N
    add  N to CLOSED
    FOR all children M of N not in CLOSED:
        M.parent = N
        M.g = N.g + cost(N,M)
        M.h = heuristic(M)
        add M to FRONTIER
    ENDFOR
ENDWHILE
```

Figure 2 .Pseudocode for A algorithm.*

ANSWER to PROBLEM (A* alg.)

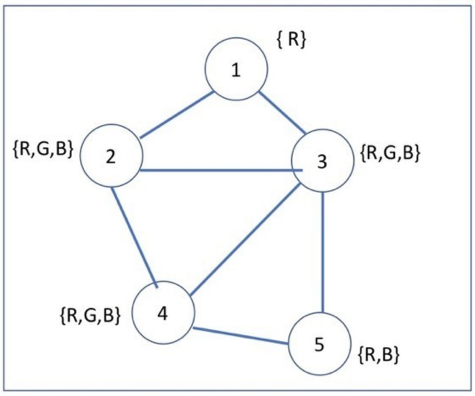
1. Expanded nodes: ACBDG.

2. Returned path: ACG. Not optimal. Because of B is not consistent. H(B) > h(C) + c(B,C) C is expanded with path cost 6, but then it appears as the child of B again with a path cost 5 (3+2) this time but since it is in "closed" is not re-expanded.
3. The pseudocode is changed so that a node in "closed" can be re-expanded (by putting it back into "frontier". This can be done by removing the "not in CLOSED" part of Sentence 9. Alternatively, the path cost (M.g+M.h) of the new encounter of the node (M) is compared with path cost of M in the "closed" and it is put back to frontier only if its new path cost is smaller.
4. No. See the example below (cost 1 versus -99)



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,

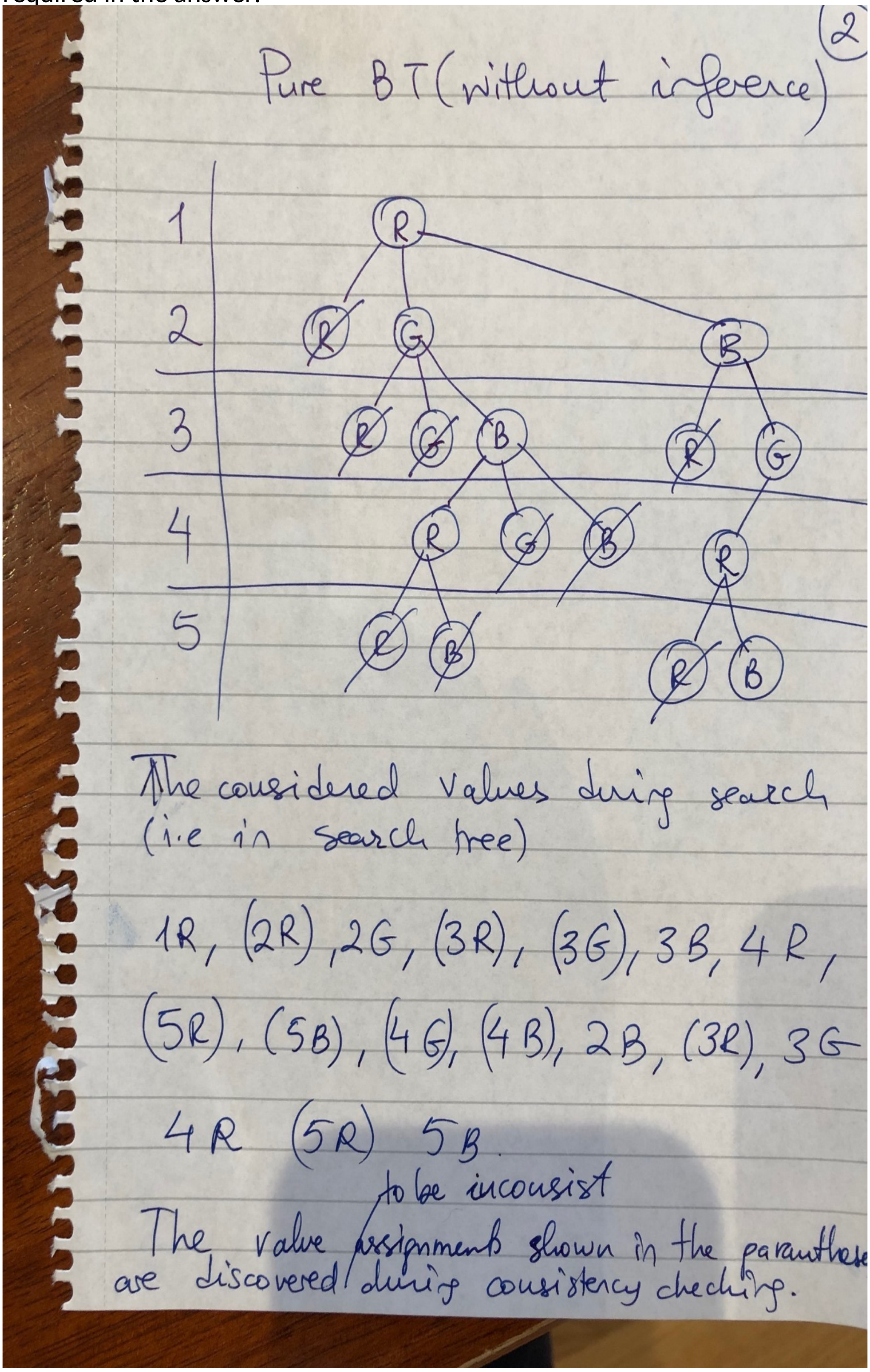
show assignments by writing the number of variables followed by the letter for the value.

ANSWER to PROBLEM (Constaint Satisfaction)

1. 1={R}
   2={G,B}
   3={G,B}
   4={R,G,B}
   5={R,B}

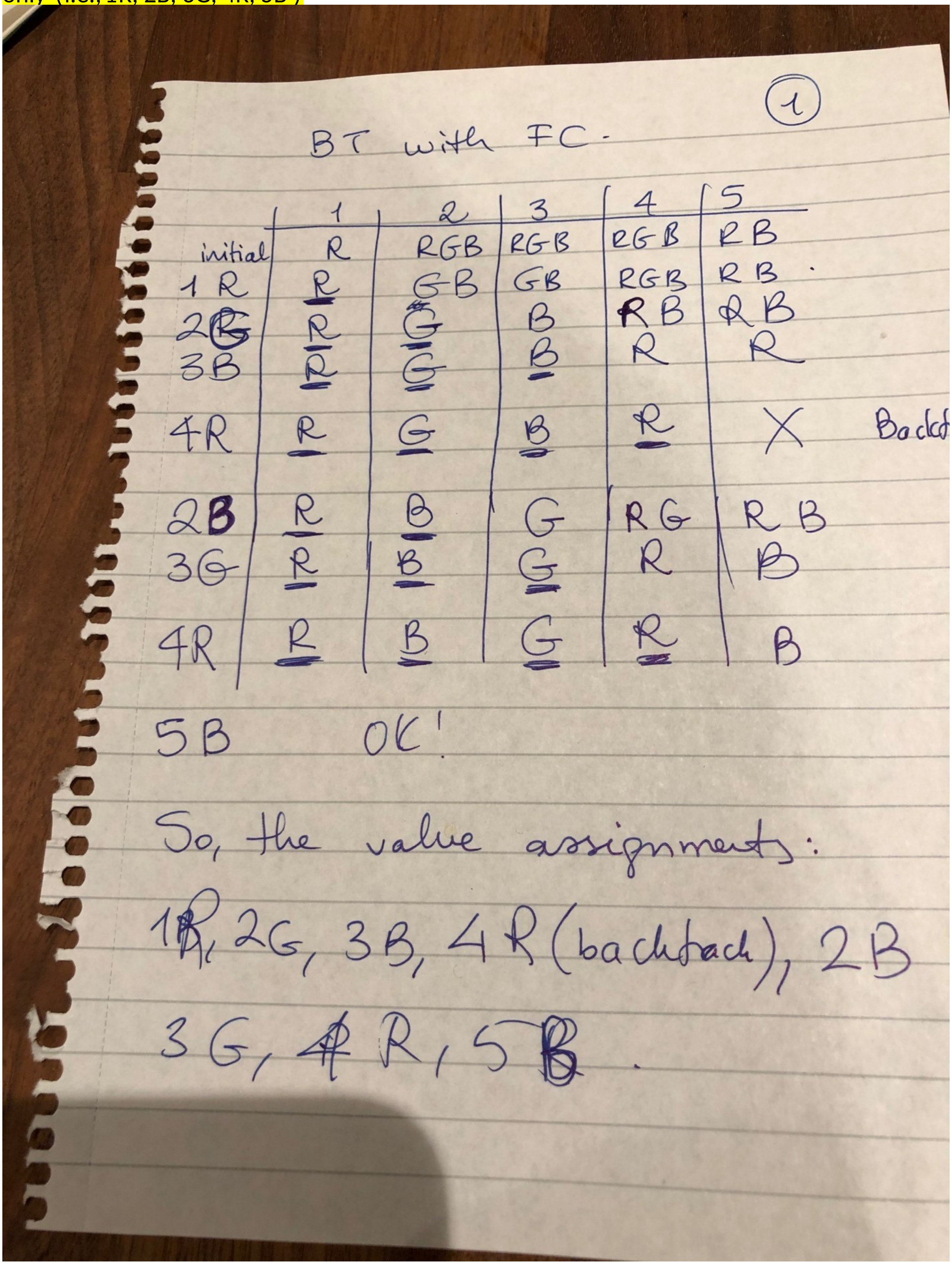
2. 1R, 2R, 2G, 3R,3G, 3B, 4R, 5R, 5B, 4G, 4B, 2B, 3R, 3G, 4R, 5R, 5B. The tried but not assigned values during backtracking search are shown in parentheses: 1R, (2R), 2G, (3R), (3G), 3B, 4R, (5R), (5B), (4G), (4B), 2B, (3R), 3G, 4R, (5R), 5B. The figure below shows all value considerations. Full 4 points have been given also to answers that don't include the inconsistent values (i.e., the ones in the parenthesis), i.e., 1R, 2G, 3B, 4R, 2B, 3G, 4R, 5B is also accepted as correct answer. So both 1R, 2R, 2G, 3R,3G, 3B, 4R, 5R, 5B, 4G, 4B, 2B, 3R, 3G, 4R, 5R, 5B . and 1R, 2G, 3B, 4R, 2B, 3G, 4R, 5B are accepted as correct answer.

The following and the next figure are in order to show/explain you, they are not required in the answer.



3. Forward checking does check only 1 step(immediate neighbours not assigned yet) after assignment of a variable

4. 1R, 2G, 3B, 4R, 2B, 3G, 4R, 5 B. No point is given if the answer is the result path only (i.e., 1R, 2B, 3G, 4R, 5B)



BT with FC.

1

	1	2	3	4	5	
initial	R	RGB	RGB	RGB	RB	
1 R	R	GB	GB	RGB	RB	
2G	R	G	B	RB	RB	
3B	R	G	B	R	R	
4R	R	G	B	R	X	Backt
2B	R	B	G	RG	RB	
3G	R	B	G	R	B	
4R	R	B	G	R	B	

5B OK!

So, the value assignments:

1R, 2G, 3B, 4R (backtrack), 2B
3G, 4R, 5B.

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	*agent2* NG
agent1	H	8, 0	3, 1
agent1	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.

Answer:

1. First notice, neither of Agent2's strategies are dominated since,

   u2 (H, G) = 0 < 1 = u2 (H,NG) and u2 (NH,G) = 4 > 3 = u2 (NH, NG)
   H strictly dominates NH for Agent1 since,
   u1 (H, G) = 8 > 4 = u1 (NH, G)
   and u1 (H, NG) = 3 > 2 = u1 (NH, NG) .

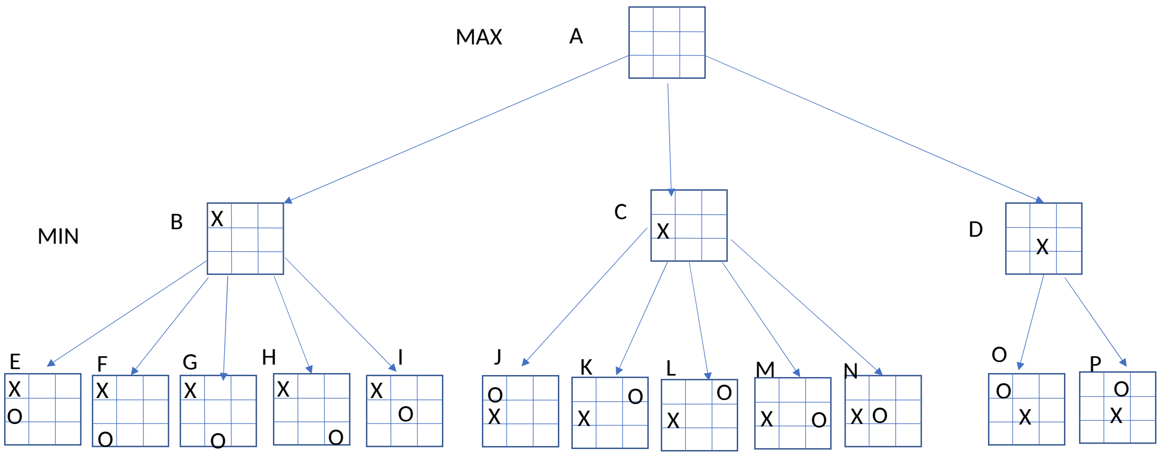
2. NE is (H, NG) which yields a payoff of (3, 1)

3. (H,G) and (NH, G) are pareto optimal

4. Social welfare maximizing profile means that the sum of utilities of all agents are highest for this action profile. This means that it is not possible to increase utilities of both agents at the same time.

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:

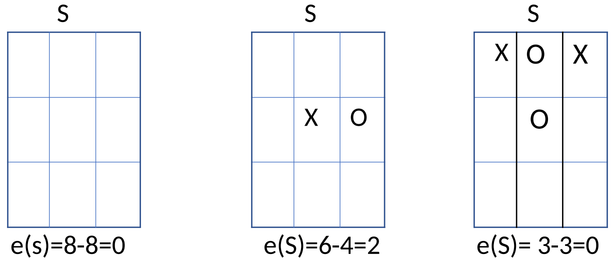


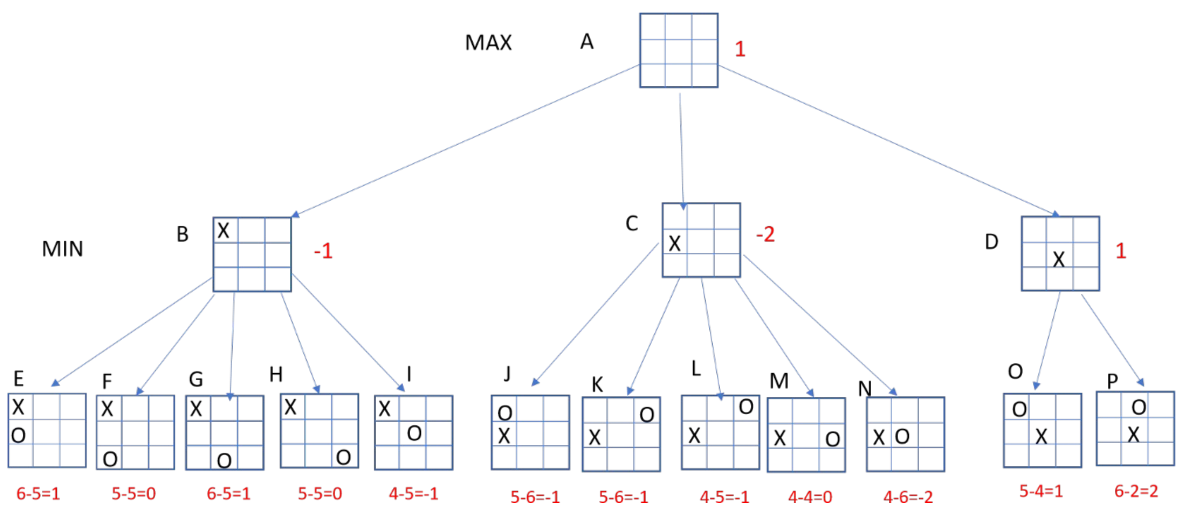
Figure : Example on estimation the values of terminal nodes

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.

ANSWER to ADVERSARIAL SEARCH

1. See figure for MINIMAX alg. Best action is the one with outcome D.
2. K,L, M,N can be pruned. Answers excluding K or excluding L are also correct since the two are equal.



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?

   Answer: the goal is implicit in the condition-action rule (goals are "designed in")

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

   ANSWER: Satisfiable. When P=T and Q=T the sentence is true, but when P=F and Q=T the sentence is false.

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

   ANSWER: ∀ x [(IsPerson(x) => ∃ y (Has(x, y) ∧ IsUmbrella(y))) => ¬ IsWet(x)] .

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

   ANSWER: ∃x, y Daughter(x, John) ∧ Daugher (y,John) ∧ ¬ (x=y)

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?

   Answer: Global minimum

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.

ANSWER: Yes, this is an admissible heuristic because it will always underestimate the distance to a solution (no solution can be nearer than the first layer where all of the solution propositions are no mutex)

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?

   ANSWER: There are no constraints on what order the actions must be executed.
   They are all in the same layer, which indicates that they can be performed in parall el.

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

   ANSWER: Backward

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?

Answer: In the "age" slot, and as a demon, a procedure/function that computes the age.

Name: student
isa: person
age: (calculate-age)

..: ..
...

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.
Answer: sequential, stochastic, 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.

ANSWER: True. Since P(S_0-> S_1) = exp($\Delta E/T$) and
ΔE =|VALUE(S_0)-VALUE(S_1)|, we see that with probab. 1 we will move to S_1.

NTNU

Department of Computer and Information Science

Examination paper for (course code) (course title)

TDT4136 - Introduction to Artificial Intelligence

Academic contact during examination: Odd Erik Gundersen

Phone: +47 47637075

Examination date: 19/12/2016

Examination time (from-to): 09:00 – 13:00

Permitted examination support material: D No
printed or handwritten material is permitted.
Calculator is permitted.

Other information:
Results: 19 January 2017
If you believe that some information is missing in the formulation of a problem, briefly describe the necessary assumptions you made.

Language: English

Number of pages (front page excluded): 6

Number of pages enclosed: 7

Checked by:

Date Signature

Students will find the examination results in Studentweb. Please contact the department if you have questions about your results. The Examinations Office will not be able to answer this.

(NOTE: All the answers need to be written down into answer sheets, not into the question list.)

TASK 1: Task Environment (10p)

Specify the task environment of the following four agents.

Note 2020: Several students have pointed out that there sometimes exists reasonable alternative answers to questions about agent environments. On future exams these kinds of questions will be discussion questions where you will explain your answers and what assumptions you make.

Possible answers:
Observable: Fully (F) or partially (P).
Agents: Single (S) or Multi (M)
Deterministic: Deterministic (D) or stochastic (S).
Episodic: Episodic (E) or sequential (S)
Static: Static (S), semi (I) or dynamic (D)
Discrete: Discrete (D) or continuous (C)

Agent 1: Deep Blue
Deep Blue is a chess playing agent that played and won against Garry Kasparov in 1997. Consider one game against Garry using a clock.

Agent 2: Roomba
Roomba is a vacuum-cleaning robot that drives around and vacuums the floors in all the rooms of a home.

Agent 3: A Tesla factory paint-robot.
A paint robot on the Tesla factory paints one and one car. The cars are transported by a transport robot to and from the paint robot. The paint robot uses a spray-painting robot arm to paint the cars.

Agent 4: Stats Monkey the robot journalist
Stats Monkey collects box scores and play-by-play data to spit out credible accounts of college baseball games while the games are being played.

Answer to be written down into answer sheet following the given table:

Task Environment	Observable P/F	Agents S/M	Deterministic D/S	Episodic E/S	Static S/D/I	Discrete D/C
Deep Blue	F	M	D	S	I	D
Roomba	P	S	S	S	D	C

Students will find the examination results in Studentweb. Please contact the department if you have questions about your results. The Examinations Office will not be able to answer this.

Paint robot	P	M	S	S	S	C
Stats Monkey	P	S	D	S	I	C

TASK 2: Propositional and first-order logic (20p)

a) (4p) What are the advantages and disadvantages of the propositional logic ? What is the difference between implication and entailment in propositional logic ? (*The answer should be shorter than one page)*.

Pros:

- Propositional logic is declarative: Pieces of syntax correspond to facts
- Propositional logic allows partial/disjunctive/negated information
- Propositional logic is compositional

Cons:

- Propositional logic has very limited expressive power.

Implication: $(S \rightarrow Q)$ is true iff $(\neg S \vee Q)$ is true.

Entailment: $(KB \models Q)$ is true iff every interpretation that makes all $(S \in KB)$, makes Q true.

b) (4p) Convert the following formula to CNF (conjunctive normal form)

$$(A \wedge B) \Rightarrow (\neg A \Leftrightarrow B)$$

$\neg A \vee \neg B$

c) (2p) Multiple choice

The formula $\forall x \exists y\, P(x,y) \rightarrow \exists q\, P(q,q)$ would be treated as a validity

1. Under all possible circumstances
2. By an inference engine that implements occurs check as Skolem functions
3. By an inference engine that implements occurs check as Skolem constants
4. Under no possible circumstances

3 (R&N, p.362)

d) (3p) True or False (*correct answer= 1p; wrong answer= -½p; total score will be 0-3p*)

1. Universal Instantiation is built on Skolemization. **F** (R&N, p.323)
2. The Backward Chaining Algorithm can be described as follows:
   i. Pose the original query as a goal.
   ii. Find every clause in the knowledge base whose right-hand side unifies with the goal under some substitution.
   iii. Prove in turn every conjunct on the left-hand sides of each of these clauses, keeping track of the accumulated substitutions. **T** (R&N, p.337)
3. There exists a sentence S in First Order Logic such that S cannot be converted into an inferentially equivalent sentence in Conjunctive Normal Form. **F** (R&N, p.345)

Students will find the examination results in Studentweb. Please contact the department if you have questions about your results. The Examinations Office will not be able to answer this.

e) The statement "Every Russian school boy knows a game"[1] has two interpretations:
A. There exists a game such that every Russian school boy knows this game.
B. For each Russian school boy, there exists a game so that the boy knows this game.

Questions:

1. (2p) Formulate each of these two interpretations in first order logic.
   Let R(x): x is a Russian schoolboy, G(y): y is a game, K(x,y): x knows y.
   A: $\exists y\ [\ G(y) \wedge (\forall x\ R(x) \Rightarrow K(x,y))\ ]$
   B: $\forall x\ [\ R(x) \Rightarrow \exists y\ (G(y) \wedge K(x,y))\ ]$

2. (2p) Convert the formulas into clausal form.
   Clausify A (eliminate $\exists$ and replace $A \Rightarrow B$ with $\neg A \vee B$):
   A: $\exists y\ [\ G(y) \wedge (\forall x\ R(x) \Rightarrow K(x,y))\ ]$
   $\exists y\ [\ G(y) \wedge (\forall x\ \neg R(x) \vee K(x,y))\ ]$
   $\exists y\ [\ G(y) \wedge (\ \neg R(x) \vee K(x,y))\ ]$
   $G(S1) \wedge (\ \neg R(x) \vee K(x,S1))\ ]$

   P1: G(s1)
   P2: $\neg R(x) \vee K(x,s1)$
   Now we convert B to CNF but we do also its negation as a preparation for the next task which is about resolution refutation

   Negate B and clausify (move $\neg$ inwards, using that $\neg\forall x\ P \equiv \exists x\ \neg P$ and $\neg\exists x\ P \equiv \forall x\ \neg P$ and De Morgan's Laws: $\neg(A \wedge B) \equiv \neg A \vee \neg B$ and $\neg(A \vee B) \equiv \neg A \wedge \neg B$):

   $\neg\forall x\ [\ R(x) \Rightarrow \exists y\ (G(y) \wedge K(x,y))\ ]$
   $\exists x\ \neg[\ R(x) \Rightarrow \exists y\ (G(y) \wedge K(x,y))\ ]$
   $\exists x\ \neg[\ \neg R(x) \vee \exists y\ (G(y) \wedge K(x,y))\ ]$

   $\exists x\ [\ R(x) \wedge \neg\exists y\ (G(y) \wedge K(x,y))\ ]$
   $\exists x\ [\ R(x) \wedge \forall y\ \neg(G(y) \wedge K(x,y))\ ]$
   $\exists x\ [\ R(x) \wedge \forall y\ (\neg G(y) \vee \neg K(x,y))\ ]$
   $R(s2) \wedge \forall y\ (\neg G(y) \vee \neg K(s2,y))$

   P3: R(s2)
   P4: $\neg G(y) \vee \neg K(s2,y)$

3. (3p) Use either a resolution proof or the Tableaux method to show that the logical formulation of interpretation A implies the logical formulation of interpretation B.

Resolution proof (use the clausal forms from task 2e2 or derive them here):
P5: $\neg K(s2,s1)$ (P1,P4). -here you do substitution y/S1

1 *The saying "Every Russian schoolboy knows …" (that you must recapture with the pawn!) is attributed to the Soviet chess Grandmaster David Bronstein who used it to imply how little Western chess players (in the 1950s) understood of the game compared to any Russian. However, Bronstein never met Magnus Carlsen* ^^

Students will find the examination results in Studentweb. Please contact the department if you have questions about your results. The Examinations Office will not be able to answer this.

P6: K(s2,s1) (P3,P2)
P7: [] (P5,P6). Empty set meaning that NEG B is inconsent with A, meaning B is implied by A.

TASK 3: Search (20p)

You are going to evaluate search algorithms that can find the shortest indoor walking paths. Figure 2 illustrates seven rooms and the actual walking distances between them. Table 1 specifies the straight-line distances between room 127D and all the other rooms. The evaluation function f(n) evaluates node n. When we evaluate the algorithms, we start our path search in room 181 and we want to find the shortest path to room 127D, which is our end node.

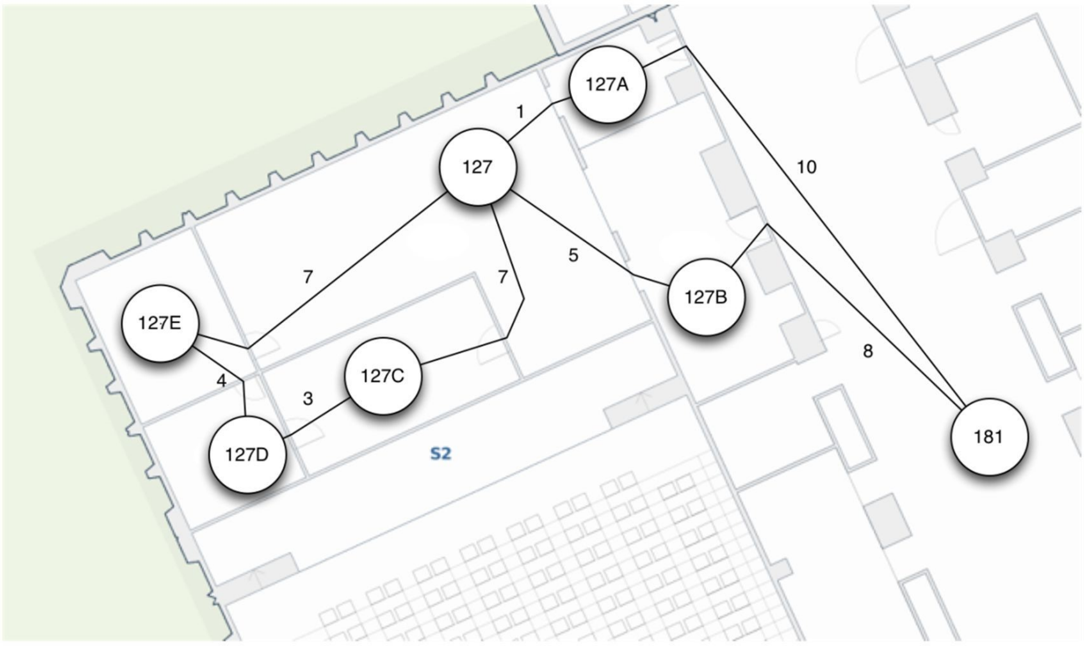


Figure 1: The graph represents the possible walking paths between the seven rooms 127D, 127E, 127C, 127, 127A, 127B and 181. The numbers close to the paths indicate the actual walking distances between the connected rooms.

Table 1: Straight line distance between the rooms and 127D

Room	SLD from 127D
127	7
127A	10
127B	9
127C	3
127E	2
181	14

a) Greedy best-first search (3p): What is the evaluation function f(n) for greedy best-first search? Write the function and describe the term(s) on the right hand side.

Students will find the examination results in Studentweb. Please contact the department if you have questions about your results. The Examinations Office will not be able to answer this.

Answer: f(n) = h(n), h(n) is a heuristic function that estimates how fra n is from the goal state, for example SLD for the indoor paths in this task.

b) A* (3p): What is the evaluation function for A*? Write the function and describe the term(s) on the right hand side.

Answer: f(n) = g(n) + h(n), h(n) is the heuristic, such as SLD, and g(n) is the actual cost of getting to n, for instance the actual distance travelled in this task.

c) Greedy best first search (3p): For each step in the search, write the evaluation function for the node that is selected for expansion.

Step	Node	f(n) = g(n)	Open/Closed
1	181	14	O: 127B (9), 127A (10) C: 181
2	127B	9	O: 127 (7), 127A (10) C: 181, 127B (9)
3	127	7	O: 127E (2), 127C (3), 127A (10) C: 181, 127B (9), 127 (7)
4	127E	2	O: 127D (0), 127C (3), 127A (10) C: 181, 127B (9), 127 (7), 127E (2)
5	127D	0	O: 127C (3), 127A (10) C: 181, 127B (9), 127 (7), 127E (2), 127D (0)

Answer: f(127B)=9, f(127)= 7, f(127E)=2, f(127D)=0. Total actual distance travelled: 24.

Students will find the examination results in Studentweb. Please contact the department if you have questions about your results. The Examinations Office will not be able to answer this.

d) A* (3p): For each step in the search, write the evaluation function including all the terms and their values for the node that is selected for expansion.

Answer:

Step	Node	g(n)	h(n)	f(n)	Frontier/Expanded
1	181	0	14	14	F: 127B (17), 127A (20) E: 181
2	127B	8	9	17	F: 127A (20), 127 (20) E: 181, 127B What if 127 is explored first (look at 3, alt, which will take one extra step to go to 3)
3	127A	10	10	20	F: 127 (18) ~~127 (20)~~, E: 181, 127B, 127A
3, alt	127	13	7	20	F: 127A (20), 127E (22), 127C (23) E:181, 127B, 127 (20) Ends up at 3.
4	127	11	7	18	F: 127E (20), 127C (21) E: 181, 127B, 127A, 127
5	127E	18	2	20	F: 127C (21), 127D (22), E: 181, 127B, 127A, 127, 127E
6	127C	18	3	21	F: 127D (20), ~~127D (22)~~, E: 181, 127B, 127A, 127, 127E, 127C (21),
7	127D	20	0	20	F: E: 181, 127B, 127A, 127, 127E, 127C (21), 127D (20)

e) Admissible heuristics (4p): Explain what an admissible heuristics is in one sentence only. Give two examples of admissible heuristics for the 8-puzzle where the objective is to slide tiles horizontally or vertically into the open space until the goal state is reached.

Answer: An admissible heuristic is one that never overestimates the cost to reach the goal. Two admissible heuristics for the 8-puzzle:

1. Number of misplaced tiles and
2. The sum of the distances that of the tiles from their goal position.

f) Optimality of A* for graph search (1p): Must the heuristics be both admissible and consistent in order for the Russel and Norvig version of A* to be optimal when applied to graph search? Alternatives: Yes or No.

Students will find the examination results in Studentweb. Please contact the department if you have questions about your results. The Examinations Office will not be able to answer this.

Answer: yes

g) Search algorithms (3p): Which of the search algorithms 1) A*, 2) genetic algorithms, 3) minimax, 4) constraint propagation should be chosen for the following search problems:
   1. Search for a schedule of flights that has some restrictions.
   2. Search for best action in backgammon.
   3. Find the best design of a car.
   4. Find the shortest route for a self-diving car.

Answer:

Task	Answer
A	4 (constraint propagation)
B	3 (minimax)
C	2 (genetic algoritms)
D	1 (A*)

TASK 4: Constraint satisfaction (20p)

Figure 2 shows the water regions in Norway, which there are eleven of, and these are Finnmark (F), Troms (T), Norland (N), Sør-Trøndelag (ST), Møre og Romsdal (MR), Østfold (Ø), Sogn of Fjordane (SF), Hordaland (H), Buskerud (B), Rogaland (R) and Vest-Agder (VA).

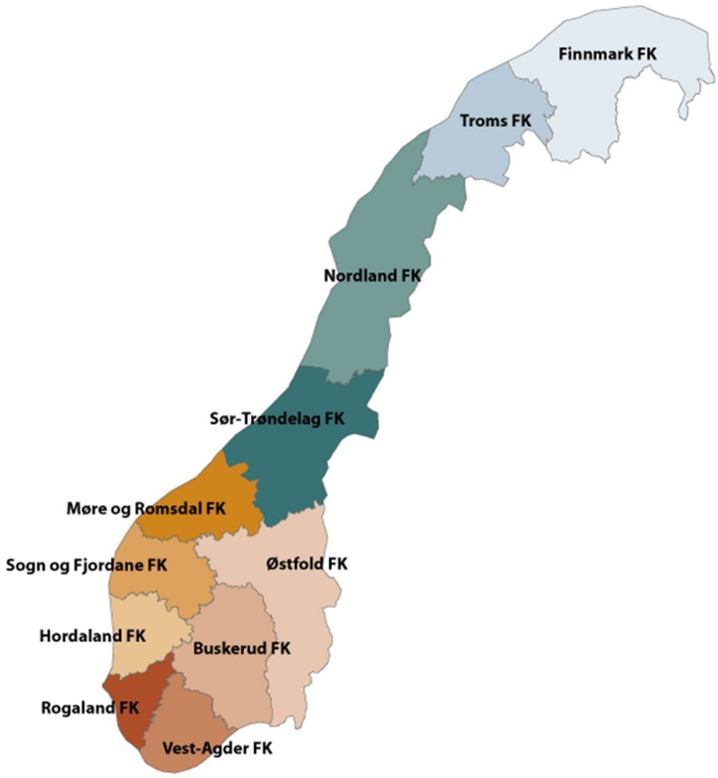


Figure 2: Water regions of Norway.

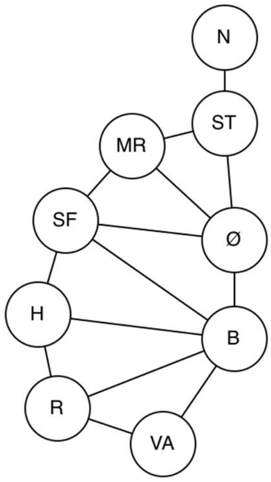
You are going to color the map. Unlike the artist that has colored the map of figure 2, you can use three colors only (red, green and blue), but no neighboring regions can have the same color. To solve this problem, you have to use your knowledge of constraint satisfaction. You will use the full constraint graph for b, and the reduced constraint graph for d and e. The reduced constraint graph only includes the regions south of Møre og Romsdal (that is we are not including Møre and Romsdal in the reduced constraint graph).

a) Constraint satisfaction problems (3p): Specify 1) the variables, 2) the domain and 3) give at least three examples of constraints.

Answer:
Variables: F, T, N, ST, MR, Ø, SF, H, B, R and VA.
Domain: red, green, blue.
Constraints: No neigboring region should have the same color. F!=T, T!=N, N!=ST.

b) Graph (3p): Draw the full constraint graph illustrating the water regions of Norway. Use the abbreviations in your graph: F, T, N and so on.

Students will find the examination results in Studentweb. Please contact the department if you have questions about your results. The Examinations Office will not be able to answer this.



c) Description (3p): Which of the following terms describe the domain and the constraints of the specified problem?

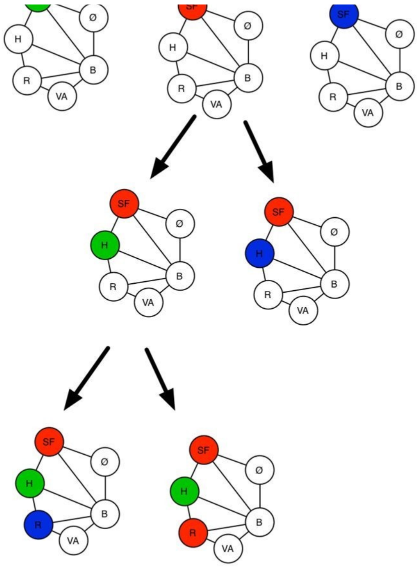
1. Discrete domain,
2. Continuous domain,
3. Finite domain,
4. Infinite domain,
5. Linear constraints
6. Nonlinear constraints,
7. Unary constraints,
8. Binary constraints,
9. N-ary constraints.

Answer:
a. Discrete domain
c. Finite Domain
h. Binary constraints

d) Backtracking search (4p): Illustrate the first three levels of the search tree of backtracking search using the constraint graph. Each node in the tree should list all the assignments made by that point in the search. Use the reduced constraint graph. You should assign values to nodes in the following order: *SF, H, R, VA, Ø*.

Answer: From book: The term backtracking search is used for a depth-first search that chooses values for one variable at a time and backtracks when a variable has no legal values left to assign.

Students will find the examination results in Studentweb. Please contact the department if you have questions about your results. The Examinations Office will not be able to answer this.



e) Forward checking (4p): Illustrate forward checking with a table. Each variable should have a column in the table, showing the remaining domain for that variable at each point in the search (represented by rows). Show three steps. Use the reduced constraint graph.

Answer:
From Book: One of the simplest forms of inference is called forward checking. Whenever a variable *X* is assigned, the forward-checking process establishes arc consistency for it: for each unassigned variable V that is connected to X by a constraint, delete from V's domain any value that is inconsistent with the value chosen for *X*.

Domain	*SF*	*Ø*	*H*	*B*	*R*	*VA*
Initial	*R G B*	*R G B*	*R G B*	*R G B*	*R G B*	*R G B*
After SF=red	*R*	*G B*	*G B*	*G B*	*R G B*	*R G B*
After H=green	*R*	*G B*	*G*	*B*	*R B*	*R G B*

Students will find the examination results in Studentweb. Please contact the department if you have questions about your results. The Examinations Office will not be able to answer this.

After R=blue	*R*	*G B*	*G*	*-*	*B*	*R G*

f) Heuristics (3p): Of the three heuristics minimum-remaining-values (MRV), degree (D) and least-constraining-value (LCV):
   a. Which heuristic should be used to choose which region to color next?
   b. Which heuristic should be used to decide the order to examine values?

Answer:
From Book:

a) The **minimum-remaining-values** (MRV) heuristic chooses the variable with the fewest "legal" values.
   The MRV heuristic doesn't help at all in choosing the first region to color. In this case, the **degree heuristic** comes in handy. It attempts to reduce the branching factor on future choices by selecting the variable that is involved in the largest number of constraints on other unassigned variables.

b) Once a variable has been selected, the algorithm must decide on the order in which to examine its values. For this, the **least constraining-value** heuristic can be effective in some cases. It prefers the value that rules out the fewest choices for the neighboring variables in the constraint graph.

Answer to be written down into answer sheet following the given table:

Task	**Answer**
a.	MRV, D
b.	LCV

TASK 5: Planning (10p)

a) Characterizing planning (5p): How can a planning (problem) be characterized? Name two situations when planning is useful. (*The answer should be shorter than one page)*.

Answer:
Planning is a way of problem solving that focuses on the effects of actions. It is in some ways similar to search however search is focused on sequences of actions that move between atomic state representations, while planning is focused on actions that move between factored state representations.

Situations when planning is useful:

- when addressing a new situation
- when tasks are complex
- when the environment imposes high risk/cost

Students will find the examination results in Studentweb. Please contact the department if you have questions about your results. The Examinations Office will not be able to answer this.

- when collaborating with others

b) Plan representation (5p): Explain how a plan can be represented? Give an example of withdrawing cash from an ATM in either STRIPS or PDDL. (*The answer should be shorter than one page*.)

A plan can be represented as:

- States: First-order predicates over objects describing a point in the search space of an application
- Actions:
  - Name: identifier of the action
  - Precondition: conjunction of literals describing whether the action can be taken
  - Effects: conjunction(s) describing the change
- Goals: A conjunction of literals

There are different languages that can be used, we discussed STRIPS and PDDL: buying a Christmas tree:

STRIPS:
Action (withdraw(cash),
PRECOND: At(ATM) ∧ Sells(ATM, cash, person) ∧ hasMoneyOnAccount (person)
DELETE-LIST: hasMoneyOnAccount (person)
ADD-LIST: have (cash))

PDDL:
Action (withdraw(cash),
PRECOND: At(ATM) ∧ Sells(ATM, cash, person) ∧ hasMoneyOnAccount (person)
EFFECT: -hasMoneyOnAccount (person) ∧ have (cash))

NB! - hasMoneyOnAccount(person) is the negation

There are also other ways to represent planning such as Situational Calculus.

TASK 6: Natural Language Processing (20p)

a) (6p) True or False (*correct answer= 1p; wrong answer= -½p; total score will be 0-6p)*

- Sentiment analysis is a text classification application.
  T (R&N, p.865)

- The purpose of smoothing is to avoid dramatic effects of low-frequency counts.
  T (R&N, p.863)

- The bag of words model can be seen as a simple language model.
  T (R&N, p.866)

Students will find the examination results in Studentweb. Please contact the department if you have questions about your results. The Examinations Office will not be able to answer this.

- ‘Freeze’, ‘halt’, ‘cease’ and ‘finish’ are examples of stop words.
  F (NLP lecture, slide 67)

- The task of Information Retrieval is to return the answer to a user query.
  F (R&N, p.867)

- Information Extraction systems are often based on templates
  T (R&N, p.874)

b) (2p) What are the main features of human languages that make parsing of these different from the parsing of programming languages?

Ambiguity and redundancy (NLP lecture, slide 5)
Other differences relate to that humans languages are constantly **evolving** and (to some extent) that they exhibit higher level of **context dependency**

c) (2p) Sentiment analysis of Twitter messages (tweets) faces several challenges. Give examples of at least four problems that need to be addressed.

There are a whole range of challenges, for example:
Word meaning can be domain dependent; modifers (e.g., negation) can change or reverse the meaning; figurative language can change the meaning (sarcasm, irony, humour); the very restricted length of the messages; brevity, idiomatic language, non-conventional grammar, spelling errors, hashtags, etc. make processing more difficult - as well as the same issues as in Task6b; references to other tweeters and tweets; links and pictures; mixing of languages;
and so on (NLP lecture, slide 48)

d) (4p) The documents in a collection that were returned respectively not returned by an Information Retrieval system in response to a given query were analysed for relevance and shown to be distributed as follows:

	Returned	Not returned
Relevant	60	40
Not relevant	20	180

1. What was the system’s precision?
   0.75 [(relevant returned) / (all returned) = 60 / (60+20) = 3/4] (R&N, p.869)

2. What was the system’s recall?
   0.60 [(relevant returned) / (all relevant) = 60 / (60+40) = 3/5]
   (R&N, p.869)

3. What is F_1 score?
   A way to combine precision and recall; their harmonic mean: 2PR/(P+R)
   (R&N, p.869)

Students will find the examination results in Studentweb. Please contact the department if you have questions about your results. The Examinations Office will not be able to answer this.

4. Calculate the F_1 score of the system.
   0.67 [2PR/(P+R) = 2*(3/4)*(3/5) / (3/4+3/5) = (18/20) / (27/20) = 2/3]
   (R&N, p.869)

e) (6p) Suppose you have access to the following knowledge sources:

- Dictionaries of basic word forms (lemmas) for English and Norwegian,
- Morphological inflection rules for the same two languages,
- A large set of English e-mails already classified as spam,
- A large set of Norwegian e-mails already classified as spam,
- A large set of English e-mails already classified as not being spam,
- A large set of Norwegian e-mails already classified as not being spam,
- A huge set of unclassified e-mails written in a wide range of human languages, and
- A stream of incoming messages, each of the length of no more than one sentence.

Describe how you would go about building a system which would analyze each incoming message and produce one of the following outputs:

1. The message is not written in a human language.
2. The message is written in a human language which is neither English nor Norwegian.
3. The message is written in English and is spam.
4. The message is written in Norwegian and is spam.
5. The message is written in English and is not spam.
6. The message is written in Norwegian and is not spam.

You do not need to produce a complete solution, but rather sketch the steps that would have to be taken.

Several solutions are possible. We could, for example, view it as a **language modelling and classification** problem, and do the following:

1. Preprocess the words in the classified and unclassified e-mails, for example by
   a. removing punctuation
   b. converting all upper case characters to lower case
2. Further analyse / normalise the words in the pre-classified e-mails:
   a. Apply the inflection rules to each word
   b. Map each word to its basic form (lemma) in the dictionaries
3. Create five language models (e.g., bag of words or bigrams) from all the words:
   a. English spam
   b. English non-spam
   c. Norwegian spam
   d. Norwegian non-spam
   e. Human language
4. Analyse an incoming message to see which language model it fits closest to
   a. by either using statistical information from the language models, or
   b. by applying a machine learning-based classifier trained on the data.

Students will find the examination results in Studentweb. Please contact the department if you have questions about your results. The Examinations Office will not be able to answer this.

Alternatively, we could address it as an **information retrieval** task (potentially after doing the same preprocessing and word-level analysis as in steps 1 and 2 in the solution above) and then:

1. Treat each incoming message as a query to the documents (the analysed emails)
2. Retrieve the top *N* (e.g., 100) documents closest matching that query.
3. Assign the class to the query which the majority of the documents belong to.
4. (If the query matches several classes, more documents can be retrieved.)

Students will find the examination results in Studentweb. Please contact the department if you have questions about your results. The Examinations Office will not be able to answer this.

Norwegian University of Science and Technology
Department of Computer and Information Science

NOTE: All the answers need to be written down into answer sheets, not into the question sheet.

Problem 1 (10 points, 2.5 pts each question))

You are a map-coloring robot assigned to the task of coloring the following map (see Figure 1). Each region must be colored one of Red (R), Green(G) or Blue(B). Adjacent regions must be a different color. The map (left) and the constraint graph (right) are shown below.

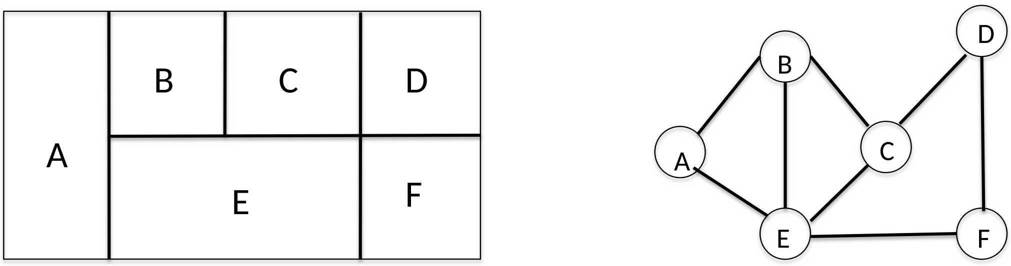


Figure 1: Map coloring problem

a) Consider the partial assignment in Figure 2 below where variable A and E have been assigned values as shown in the figure. Cross out all values that should be eliminated by Arc Consistency (using AC-3 in the textbook.)

Regions in the map

A	B	E	C	D	F
Blue	R G B	RED	R G B	R G B	R G B

Figure 2: Partial assignment for Region A= Blue and E= Red

ANSWER: See Figure 3

Regions in the map

A	B	E	C	D	F
Blue	R G B X X	Red	R G B X X	R G B X	R G B X

Figure 3: Answer to (a)

Figure 3: Answer to (a)

b) Minimum Remaining Values Heuristics. Consider the partial assignment in Figure 4 where B is assigned value Green and constraint propagation has been done. Write down all unassigned variables (just the letters that correspond to these variables) that might be selected by Minimum Remaining Values Heuristic (MRV)

Region in the map

Regions in the map

A	B	E	C	D	F
R B	Green	R B	R B	R G B	R G B

Figure 4: Partial assignment for Region B= Green

Answer: A, E, C

c) Degree Heuristic. Consider the partial assignment in Figure 4 where B is assigned Green and constraint propagation has been done. Write down all unassigned variables that might be selected by the Degree Heuristics

Answer: E

d) Consider the following complete but inconsistent assignment in Figure 6. E has been selected to be assigned a value during local search for a complete and consistent assignment. What new value would be chosen for E by the MINIMUM-CONFLICT heuristic?

Regions in the map

A	B	E	C	D	F
R B	Green	R B	R B	R G B	R G B

Figure 5: Partial assignment for Region B= Green

Regions in the map

A	B	E	C	D	F
B	G	?	G	G	G

Figure 6: Complete assignment where the value of Region E needs to be changed, and to be found.

Answer: R

Problem 2 (10 points)

Assume the following knowledge base KB:

$\forall$ x allergies(x) $\Longrightarrow$ sneeze(x)

$\forall$ x $\forall$ y cat(y) $\wedge$ allergicToCats(x) $\Longrightarrow$ allergies(x)

cat(Felix)

allergicToCats(Mary)

The Goal/Query: Does Mary sneeze?, I.e., sneeze(Mary)?

Perform Resolution Refutation (RR) and find out if the query has a positive (True) or negative (False) answer for this KB? Answer the question with True (Yes) or False (No), and show how you derived this answer through RR. Show also unifications if any. T/F answers without the poof/refutation of the query will not be given any point. Partial points will be given to the proof.

ANSWER: See (Figure 7)Result of Resolution is {}, i.e, Mary sneezes. i.e., sneeze(Mary) is true. 1 point for unification. No point reduction for lack of standardization (i.e., variable w in the figure)

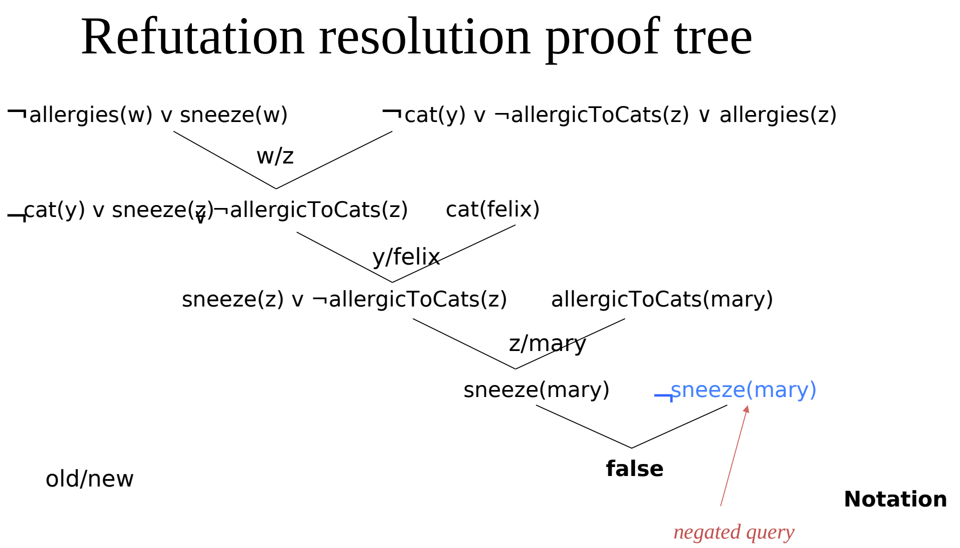


Figure 7: ANSWER to Resolution Refutation Problem.

Problem 3 (10 points, division: 3-3-4 pts)

In the following adversarial game (see Figure 8) it is MAX's turn to play. The numbers at each leaf node is the estimated score of that position. Check nodes from left to right order.

a) Perform mini-max search and label each branch node with its value. Draw the figure on your answer sheet and fill the squares above the leaf nodes. You don't need to draw the nodes at the leaf level.

ANSWER: See Figure 9

b) What is Max's best move, i.e., which node it is and which utility does it give?

ANSWER: Node B, value 5

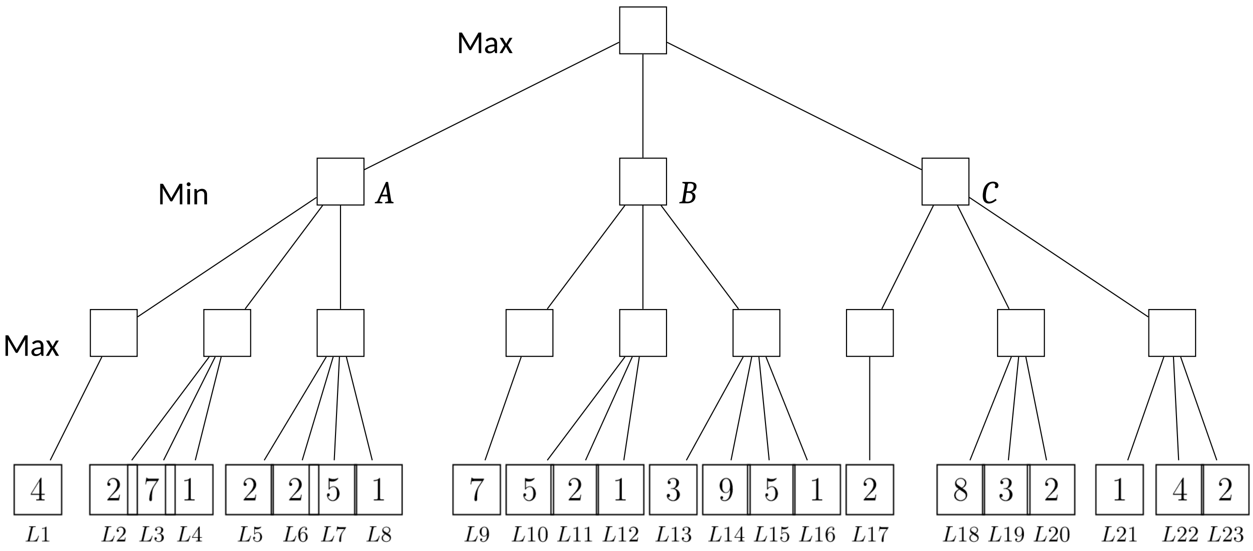


Figure 8: Adversarial game problem.

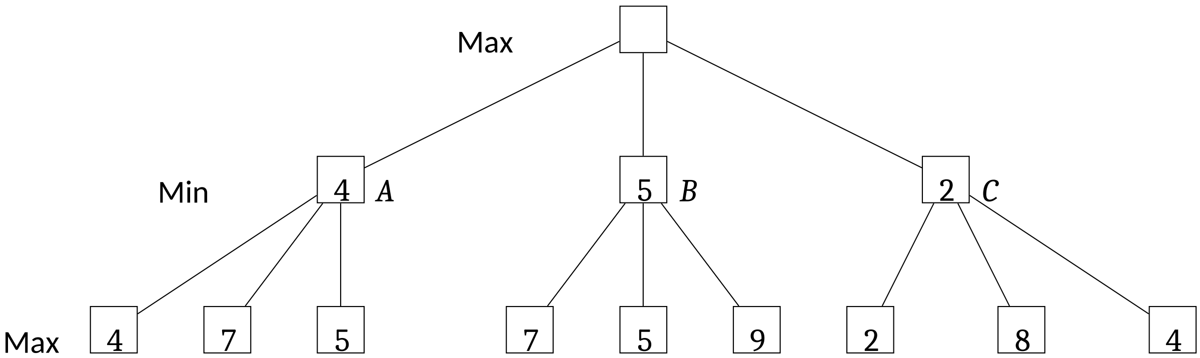


Figure 9: Minimax.

c) Write down all leaf nodes that the alpha-beta algorithm would prune. Your answer will be a list of L_i labels .

ANSWER: L4, L8, L15, L16,L18, L19,L20, L21, L22, L23)

Problem 4 (15 points total, 4 points for each of the first questions. The sum will not be negative for this problem.)

This is a mixture of multiple choice (b), Classical "writing the answer" (a and c), and True/False type of tasks.

a) Assume the following problem represented as a graph (see Figure 10) where the numbers on the edges represent the cost of traveling between the nodes connected by the edge. You will apply A* graph search algorithm and the nodes in the graph have the following h-values:

h(S)= 7
h(A)= 6
h(B)= 2
h(C)= 1
h(G)= 0

Write down the to-be expanded node and the content of the priority queue at each step. Ties break alphabetically. Write down the generated solution path - you will not get any points if you don't show how the solution is generated (i.e., the queue and the currently expanding node). Is this an optimal solution? Why or why not, explain very briefly.

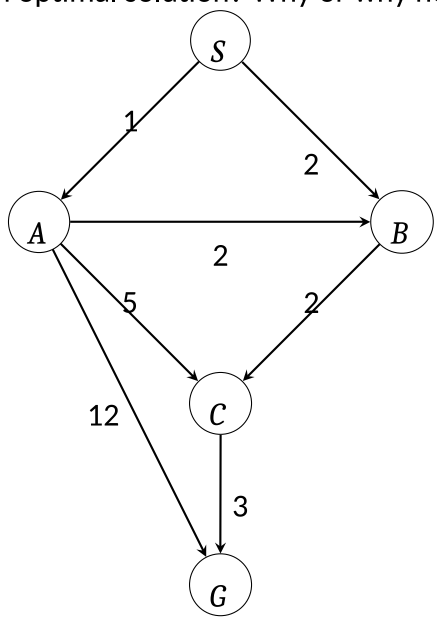


Figure 10: A* Problem

ANSWER: The solution is S B C G. Optimal. Because cost for this solution is 7 and there is no other path with less cost. SEE FIGURE below for expansion and Queue. Solution: SBCG

- <u>Expanding Queue</u>
  S(7)
- S (7) -à A(1+6,S), **B(4,S)**
- B(4,S) A(7,S), **C(5,SB)**
- C(5,SB) **A(7, S)** , G(7+0,SBC)
- A(7,S) **G(7,SBC)**
- G(7,SBC) goal

b) Which of the following set of heuristic values provide a consistent heuristic for this problem (Figure 10)? Choose one of the options below.

1. $\{h(S) = 7, h(A) = 6, h(B) = 2, h(C) = 1, h(G) = 0\}$
2. $\{h(S) = 7, h(A) = 6, h(B) = 2, h(C) = 2, h(G) = 0\}$
3. $\{h(S) = 7, h(A) = 6, h(B) = 4, h(C) = 2, h(G) = 0\}$
4. $\{h(S) = 7, h(A) = 4, h(B) = 4, h(C) = 4, h(G) = 0\}$
5. None of the above. IMPORTANT: Note that for consistency, every pair of adjacent nodes (node that have direct connection) should be checked. Not only between S–>A or A–B but all pairs, and all nodes must be used in these pairs (including G, which is also "a node". Those answers which make part of the Figure consistent but not the entire one will not be accepted as correct.

(ANSWER: 5, None of the above. For example $\{h(S) = 7, h(A) = 6, h(B) = 2, h(C) = 1, h(G) = 0\}$ is not consistent because for example h(A)-h(B)=4 is not ≤ C(A,B)=2). Remember that for consistency, h(n)-h(n') ≤ C(n, n'), should be true throughout the graph

c) Write down a consistent heuristic set and apply A* graph search with it on the same graph (Figure 10). What is the solution path with this heuristic set? You don't get any points by writing only the optimal solution if it is not accompanied with a consistent set of heuristics.

One consistent set is: $\{h(S) = 7, h(A) = 6, h(B) = 5, h(C) = 3, h(G) = 0\}$.

Check for h(n)-h(n') ≤ C(n, n') condition for consistency. Solution is SBCG.

d) Uniform cost Seach is a special case of A-star. True or False?

ANSWER: True, the case where (h = 0)

e) Simulated annealing is a stochastic optimization method. True or False?

ANSWER: True

Problem 5 (15 points. Division: 3-2-2-3-2-3, and -1 for each wrong answer. Sum of the points will not be negative for the whole problem.)

This is a combination of multiple-choice and True/False types of task. Choose the correct answer for each question.

a) "Men and women are welcome to apply." is equivalent to

$\forall(x)$ [(M(x) ∧ W(x)) $\Longrightarrow$ Apply(x)]

ANSWER: False. The FOL sentence says that everything that is both a man and a woman is welcome to apply, obviously not what is meant. The disjunction would give a correct one

b) Which of the following represents the sentence "Some person plays every game"?

1. $\exists x \forall y [Person(x) \wedge Game(y) \rightarrow Plays(x,y)]$
2. $\exists x \forall y [Person(x) \wedge Game(y) \wedge Plays(x,y)]$
3. $\forall x \forall y [Person(x) \wedge [Game(y) \rightarrow Plays(x,y)]]$
4. $\exists x \forall y [Person(x) \wedge [Game(y) \rightarrow Plays(x,y)]]$
5. None of the above.

ANSWER: 4. $\exists x \forall y [Person(x) \wedge [Game(y) \rightarrow Plays(x,y)]]$.

c) Which of the following is the Skolemized version of this sentence:

$\forall x [(\neg P(x) \wedge Q(x)) \vee \exists y (R(x,y) \wedge T(y)))]$

1. $\forall x [\neg P(x) \wedge Q(x) \vee (R(f(x),y) \wedge T(y)))]$
2. $\forall x [P(x) \wedge \neg Q(x) \vee (R(x,f(x)) \wedge T(f(x)))]$
3. $\forall x [\neg P(x) \wedge Q(x) \vee (R(x,f(x)) \wedge T(f(x)))]$
4. $\forall x [\neg P(x) \wedge Q(x) \vee (R(x,x) \wedge T(x)))]$
5. None of the above.

ANSWER: 3.

$\forall x [\neg P(x) \wedge Q(x) \vee (R(x, f(x)) \wedge T(f(x)))]$.

d) For the sentence below, write V=valid if the sentence is valid, U=unsatisfiable if the sentence is unsatisfiable and S=satisfiable if the sentence is satisfiable but not valid.

1. $\forall x[[Student(x) \wedge \neg Student(x)] \rightarrow BornOn(x,Moon)]$

ANSWER: Valid. Because the antecedent is false.

e) Apply one step resolution to the following clauses: $p \vee q$ and $\neg p \vee \neg q$. Which of the below is the correct result of the resolution step.

1. $\{p \vee \neg p\}$ and $\{q \vee \neg q\}$
2. $\{\}$

(ANSWER: $\{p \vee \neg p\}$ and $\{q \vee \neg q\}$ is correct. Because when two clauses have multiple pairs of complementary literals, only one pair of literals may be resolved at a time.)

f) You will convert the following sentence into Conjunctive Normal form.

$(A \wedge B) \vee (C \wedge D) \vee (E \Rightarrow F)$

Which one is the resultant CNF?

1. $(A \vee C) \wedge (B \vee C) \wedge (A \vee D) \wedge (B \vee D \vee \neg E \vee F)$
2. $(A \vee C) \vee (B \vee C) \vee (A \vee D) \wedge (B \vee D \vee \neg E \vee F)$
3. $(A \vee C) \wedge (B \vee C) \wedge (A \vee D) \wedge (B \vee D \vee E \vee F)$
4. $(A \vee C \vee \neg E \vee F) \wedge (A \vee D \vee \neg E \vee F) \wedge (B \vee D \vee E \vee F) \wedge (B \vee C \vee E \vee F)$
5. none of the above

ANSWER: None of the above. Correct answer: $(A \vee C \vee \neg E \vee F) \wedge (A \vee D \vee \neg E \vee F) \wedge (B \vee D \vee \neg E \vee F) \wedge (B \vee C \vee \neg E \vee F)$

Problem 6 (20 points. Each question is 2 points. -1 for each wrong answer but the total points will not be negative.)

True/False type of questions. Answer with either True(T) or False(F)

a) Randomized behaviour may be rational in competitive multi-agent environments.

ANSWER: TRUE, in competitive multi-agent environments it can avoid the pitfalls of predictability

b) There exists a task environment in which every agent is rational.

ANSWER: True. Consider a task environment in which all actions (including no action) give the same, equal reward

c) Philosopher John Searle suggests that any physical symbol system has necessary and sufficient means for general intelligent action. True or False?

ANSWER: False. This is suggested by Alan Newel and Herbert Simon. Searle suggested the opposite

d) If h is a consistent heuristic, then h is also an admissible heuristic. True or False? ANSWER: True.

e) Purely reactive agents use semantic networks for planning. True or False?

ANSWER: False.

f) Backgammon is a fully observable, sequential, deterministic, static, discrete and multiagent environment. True or False?

ANSWER: False not deterministic.

g) Circumscription allows the entailed sentences to be removed after new sentences added to the knowledge base. ANSWER: True

h) "Multiple inheritance" is one of the reasons that leads to undecidability problem in first order logic. True or False?

ANSWER: False. It is not in logic but semantic networks

i) "Closed world assumption" is the assumption that atomic sentences not known to be true are in fact false.

ANSWER: True

j) Two agents participate in a game which is defined as follows:

Agents. {Agent-i, Agent-j}

Actions: {0,100,200,300} - these actions can be thought as giving bid in an auction for example.

Agent i \ Agent j	0	100	200	300
0	**300,0**	**200,0**	**0,100**	**0,0**
100	200,0	200,0	200,100	0,300
200	0,0	0,200	0,0	0,0
300	-100,0	-100,0	-100,0	-100,0

Figure 12: Game theory question

The Payoff matrix is shown in Figure 12 .

Which of the options below is the strongly dominant equilibrium in this game? Each option shows the pair of actions, (action of agent-i, action of agent-j):

1. (0, 100)
2. (200, 200)
3. (300,0)
4. (300,100)
5. None of the above

ANSWER: None of the above. There is no strongly dominant equilibrium in this payoff matrix.

Problem 7 (20pts- division of points: 3-3-3-3-2-2-2-2. Minus point: -1 for each wrong answer)

Multiple Choice or True/False type of tasks.

a) In situation calculus, something true in one situation may not be true in another situation. True or False?

ANSWER: True

b) The following plan describes the process of withdrawing money from an ATM. Which of the representations is written in STRIPS?

1. Action (withdraw(cash),

PRECOND: At(ATM) ∧ Sells(ATM, cash, person) ∧ hasMoneyOnAccount(person)
DELETE-LIST: hasMoneyOnAccount(person)

ADD-LIST: have(cash))

2. Action (withdraw(cash),
   PRECOND: At(ATM) ∧ Sells(ATM, cash, person) ∧ hasMoneyOnAccount(person)
   EFFECT: ¬ hasMoneyOnAccount(person) ∧ have(cash))

ANSWER: 1

c) Which statements about partial order planning are true?

   1. Search in plan space and use least commitment when possible.
   2. Make only choices that are relevant to solving the current part of the problem.
   3. Both of the above.
   4. None of the above.

ANSWER: 3

d) Progression planners reason from the goal state, trying to find the actions that will lead to the start state. True or False?

ANSWER: False. That is regression planner.

e) In Medieval Europe, Latin worked as the Lingua Franca. Today English has taken over that role, obviously giving an advantage to persons who have it as mother tongue. There have been attempts to create artificial human languages (e.g., Esperanto and Interlingua) that would not give anybody such an advantage, but those languages have not been very successful. Suppose you were given the task of creating such as language. Which of the following would say say would be most important to try to restrict:

   1. The lexicon.
   2. The grammar.
   3. Ambiguity.
   4. Redundancy

ANSWER: 2 or 3

f) Two search engines were evaluated on a web search query. Engine A returned 27 web pages, 18 of which were deemed relevant to the query. Engine B returned 9 pages, all of which were relevant to the query. Which of the following statements is definitely correct?

1. System A had a higher recall than system B.
2. System B had a higher precision than system A.
3. System B had a higher F1-score than system A.
4. System A had a better performance than system B.

ANSWER: 2

g) When performing sentiment analysis, it is normally important to find out:

1. Who the opinion holder is.
2. What object the opinion is expressed on.
3. If the opinion is positive, negative or neutral.
4. All of the above.

ANSWER: 3

h) Which of the following statements is not correct?

1. The distributional hypothesis assumes that words with similar usage have similar meanings.
2. The bag of words model ignores the grammatical structure of the language.
3. An n-gram model is an example of a language model.
4. Grounding means that a speaker defines the basis for a dialogue.

ANSWER: 4

LYKKE TIL!

Norwegian University of Science and Technology
Department of Computer and Information Science

Faglig kontakt under eksamen: Pinar Øzturk: (91897451 eller 73551019)

Introduksjon til Kunstig Intelligens (TDT4136)

30th November 2015 Tid:
09:00 – 13:00

Language: English-Bokmål-Nynorsk
Aid - Tillatte hjelpemidler:

No printed or hand written material is allowed. Simple calculator is allowed. Ingen trykte eller håndskrevne hjelpemidler tillatt. Bestemt, enkel kalkulator tillatt.

If you think some necessary information is missing from a question please explain what assumptions you find it necessary to make.

Problem 1 (20 pts, 2pts each question)

Answer the following questions with TRUE or FALSE.

a) Knitting is a fully observable, episodic, stochastic, static and exciting agent environment. ANSWER: False. not episodic, not stochastic

Note 2020: Several students have pointed out that there sometimes exist reasonable alternative answers to questions about agent environments. On future exams these kinds of questions will be discussion questions where you will explain your answers and what assumptions you make.

b) Procedural attachment is used in semantic networks. ANSWER: True

c) Ontology is not a key component in simple-reflex agents. ANSWER: True.

d) *Recall* is an evaluation metric used in information retrieval that measures the proportion of returned documents that are truly relevant ANSWER: False. This is precision

e) Simulated annealing is a local search method. ANSWER: True

f) Term Frequency (TF) defines the count of a term *t* in a collection of documents. ANSWER: False. In a single document.

g) A common heuristic function for 8-puzzle game is Manhattan distance which is the sum of the distances of the tiles to their goal positions.
ANSWER: TRUE

h) If both H1 and H2 are admissible heuristics for a problem and H2<H1, then H2 is a better heuristic.
ANSWER: False

i) An agent must think like a human in order to pass the Turing test. ANSWER: False. It has to act like a human.

j) Iterative deepening search is optimal if step-costs is a constant, the search-space is finite and a goal exists. ANSWER: True.

Problem 2 (15 pts, 3 pts each question)

Choose the correct answers (one for each question) to the questions below.

a) Suppose the following action schema in a planning system for 8-puzzle.

$Action(Slide(t, s_1, s_2),$

PRECOND: $On(t, s_1) \wedge Tile(t) \wedge Blank(s_2) \wedge Adjacent(s_1, s_2)$

EFFECT: $On(t, s_2) \wedge Blank(s_1) \wedge \neg On(t, s_1) \wedge \neg Blank(s_2))$

Which of the following needs to be removed from the action schema in order to get "number-of-misplaced-tiles" heuristic?

A. $Blank(s_1)$

B. $Blank(s_2)$

C. $Adjacent(s_1,s_2)$

D. $Blank(s_2) \wedge Adjacent(s_1,s_2)$

E. None of the above

ANSWER: d – $Blank(s_2) \wedge Adjacent(s_1,s_2)$.

b) We look at a Constraint Satisfaction Problem (CSP) with the three variables X, Y and Z. Let the domain for each of these variables be the set of integers from 1 to 3:

$D_x = D_Y = D_Z = \{1,2,3\}$

Let the following binary constraint $C_{X,Y}$ apply between X and Y, and $C_{Y,Z}$ between Y and Z:

$C_{X,Y} = [(1,1),(2,1),(2,2),(3,1),(3,2),(3,3)]$

$C_{Y,Z} = [(2,1),(3,1),(3,2)]$

After running AC-3, what is the domain of X?

A. $\{1,2,3\}$ D. $\{1,2\}$ G. $\{3\}$

B. $\{2,3\}$ E. $\{1\}$

C. $\{1,3\}$ F. $\{2\}$ H. $\{\}$

ANSWER: B

As described in Chapter 6.1 in the textbook (7.1 in the green version), a constraint specifies the *valid* combinations of values between two given variables. Given that there is a constraint between two variables, any combination of values that is not explicitly listed in the constraint will thus not be valid. Therefore, according to $C_{X,Y}$ above, X has the following valid values:

- 1, if Y is 1
- 2, if Y is 1 or 2
- 3, if Y is 1, 2 or 3

Y's domain is currently t1,2,3u, so we cannot yet remove any values from X's domain. However, we then move on to $C_{Y,Z}$ and see if any values can be removed from Y's domain. According to $C_{Y,Z}$, Y has the following valid values:

- 2, if Z is 1

- 3, if Z is 1 or 2

Hence, 1 is not a valid value for Y's domain, so the domain is now reduced to {2,3}. A crucial point is that we now need to check $C_{X,Y}$ again, to see if there are any consequences for X's domain from the changes in Y's domain. Indeed there is; with 1 removed from Y's domain, 1 is no longer a valid value for X either. X's domain is thus {2,3}.

c) Z is then set to 2 in the CSP from the previous question, and AC-3 is run again. What is now the domain of X?

A. {1,2,3}
B. {2,3}
C. {1,3}
D. {1,2}
E. {1}
F. {2}
G. {3}
H. { }

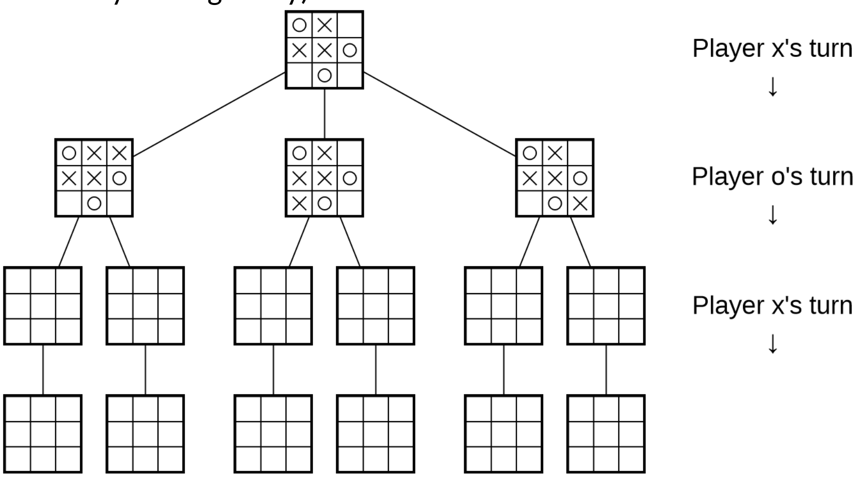
ANSWER: G

Recall that, according to $C_{Y,Z}$, Y has the following valid values:

- 2, if Z is 1
- 3, if Z is 1 or 2

Z is 2, so the only valid value for Y is 3. According to $C_{X,Y}$, then 3 is also the only valid value for X.

d) Consider the following (incomplete) game tree for tic-tac-toe. Tic-tac-toe is a two-player game where players "x" and "o" take alternating turns to place their respective symbols in an empty cell on the 3 x 3 game board, with the goal of getting three in a row (horizontally, vertically or diagonally):



Which one of the following statements is true?

A. Player x is guaranteed to win

B. Player o is guaranteed to win

C. Player x can win, but only if player o plays suboptimally

D. Player o can win, but only if player x plays suboptimally

E. Either player can win, if the other player plays suboptimally F. Neither/none of the players can win

ANSWER: C

e) Which of the following may be most useful in generation of a semantic network automatically from documents?

A. Information retrieval

B. Sentiment analysis

C. Syntactic parsing

D. Information extraction

E. none of the above.

ANSWER: D. information extraction.

Problem 3 (15 points)

a) (2 points) For the following sentence in English, is the accompanying sentence in first-order logic a good translation? If yes, answer yes. If no, explain why not.

No two NTNU students have the same ID number.

$\neg\exists x,y,z(NTNUStudent(x) \wedge NTNUStudent(y) \wedge \neg (x = y)) \Rightarrow (IDNum(x,z) \wedge IDNum(y,z))$

ANSWER: This is NOT correct because it uses => instead of $\wedge$ when quantified by $\exists$.

CORRECT: $\neg\exists x,y,z NTNUStudent(x) \wedge NTNUStudent(y) \wedge \neg (x = y) \wedge (IDNum(x,z) \wedge IDNum(y,z))$

b) (2 points) For the following sentence in English, is the accompanying sentence in first-order logic a good translation? If yes, answer yes. If no, explain why not.

All mammals except whales are similar to humans.
$\forall x,y Mammal(x) \wedge \neg Whale(x) => Mammal(y) \wedge Human(y) \wedge Similar(x,y)$

ANSWER: This is NOT correct because it says that if there is at least one mammal that is not a whale, then every mammal has to be a human.

CORRECT: $\forall x,y Mammal(x) \wedge \neg Whale(x) \wedge Mammal(y) \wedge Human(y) => Similar(x,y)$

c) (2 points) Consider the following knowledge base containing four sentences in propositional logic:

$A => (B \vee C)$

$\neg \mathrm{A} => (B \vee C)$

$\neg C$

$(B \vee D) => E$

Can these four sentences be converted to a set of Horn clauses? If yes, write them down; if not, explain why not.

ANSWER: No because the first sentence has two literals on the right hand side, meaning that the CNF has two positive literals in it.

d) (2 points) Consider the following knowledge base containing four sentences in propositional logic: $A => (B \vee C)$, $\neg A => (B \vee C)$, $\neg C$, $(B \vee D) => E$

Convert the four sentences above into conjunctive normal form(CNF) and show the result as a set of clauses.

ANSWER:

$\neg \mathrm{A} \vee B \vee C$, $A \vee B \vee C$, $\neg C$, $\neg B \vee E$, $\neg D \vee E$

e) (3 points) Is the following sentence (1) the correct skolemization (i.e., elimination of existential quantificator) of the sentence $\forall x Person(x) \Longleftrightarrow \exists y Heart(y) \wedge Has(x,y)$?

(1) $\forall x\ Person(x)$ => $Heart(H1) \wedge Has(x,H1)$.

Why not? Write down the correct one.

ANSWER: No because this means that everone has the same heart called H1. Heart should be a function of a person.

CORRECT: (1)$\forall x Person(x)$ => $Heart(H(x)) \wedge Has(x,H(x))$.

f) (4 points) Suppose the following facts are in the knowledge base:

- Pia works in a restaurant $R(Pia)$
- Georg works in a restaurant $R(Georg)$
- Anyone who works in a restaurant and makes a big mistake is fired
  $\forall x R(x) \wedge M(x)$ => $F(x)$
- The restaurant owner is happy with anyone who doesn't make a big mistake
  $\forall y\ \neg M(y)$ => $H(owner,y)$
- Anyone who is happy with Pia is unhappy with Georg
  $\forall w H(w,Pia)$ => $\neg H(w,Georg)$

Using resolution refutation, prove that "there exists someone who makes a big mistake and is fired". Show your proof on a tree starting from the boxes in the figure below. Copy the boxes into your answer sheet and fill in the last box. Apply resolution and clearly indicate the clauses being resolved in each step. Show also the binding of variables in each step/link.

ANSWER:

convert to CNF:

1. $R(Pia)$
2. $R(Georg)$

3. $\forall x(R(x) \land M(x) => F(x))$

   $\neg R(x) \lor \neg M(x) \lor F(x)$
4. $\forall y(M(y) => H(owner, y))$

   $\neg M(y) \lor H(owner, y)$
5. $\forall w(H(w, Pia) => \neg H(w, Georg))$

   $\neg H(w, Pia) \lor \neg H(w, Georg)$

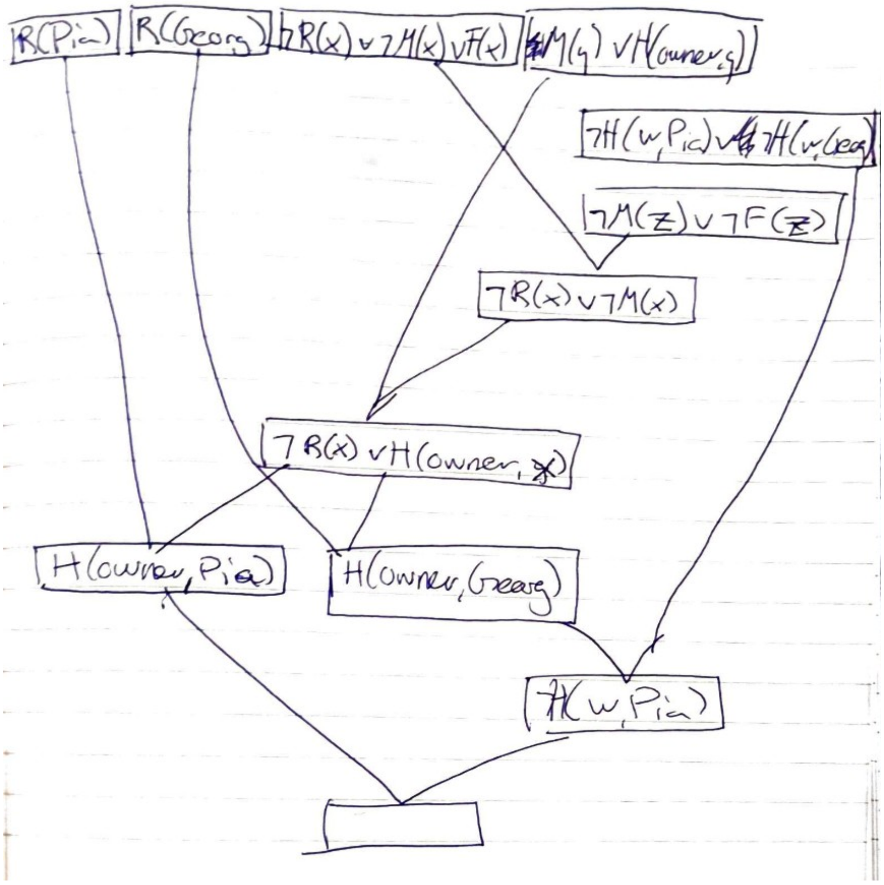
"There exists someone who makes a big mistakes and is fired"

$\exists z(M(z) \land F(x))$

Negated:

$\forall z(\neg M(z) \lor \neg F(x))$

$\neg M(z) \lor \neg F(x)$



Problem 4 (10 points)

NOTE: There is a correction in the points of the sub-questions. (a) should be 4 points and (b) is 2 points.

We changed the rules of the wumpus world. First of all, we are dealing only with wumpuses, not with gold, breeze, stench, etc. There may be more than one wumpus in the grid and they may be in any square. In the beginning of the game all squares are blank and the agent does not know which square(s) contains a wumpus. When the agent clicks on a square with a wumpus, she loses the game. If the square that the agent clicked on does not contain a wumpus, a number will appear on that square indicating the number of wumpuses adjacent to the square. Adjacency here means the four squares to the immediate left, right, top and bottom, excluding the diagonals. The goal of the game is to click on all squares which do not have wumpuses. Suppose you are playing the game in the following figure showing the current state of the grid. In the figure, squares marked *a*, *b*, *c*, and *d* are not clicked on yet. You may refer to these squares by these variables. The upper-left square is labeled 1, meaning that it is adjacent to exactly one wumpus. The other squares show the number of wumpuses in their adjacent squares.

1	a
b	2
d	c

a) (4 points) Represent the current state of the grid using propositional logic. Use the predicate W(s) to express that square *s* contains a wumpus, and $\neg W(s)$ *s* not having a wumpus.

b) (2 points) Suppose you click on the lower-left square (which is a *d* in the beginning, see the figure) and number two (2) is revealed. This means that there is no wumpus in that square but two of squares adjacent to it have wumpuses. Represent this situation of the grid using propositional logic.

c) (4 points) Prove $\neg W(a)$, that is, the square labeled *a* does not contain a wumpus. For this use the combination of initial knowledge base and the new knowledge obtained by clicking on *d*. Show your proof by resolution refutation on a drawing.

ANSWER:

a) First part (upper left is 1):

$(W(a) \wedge \neg W(b)) \vee (W(b) \wedge \neg W(a))$

Or one of the equivalent expressions:

- $W(a) <=> \neg W(b)$
- $(W(a) => \neg W(b)) \wedge (\neg W(b) => W(a))$
- $(W(a) \vee W(b)) \wedge (\neg W(a) \vee \neg W(b))$
- $(W(a) \vee W(b)) \wedge \neg (W(a) \wedge W(b))$
- …

Second part (middle right is 2):

$(W(a) \land W(b) \land \neg W(c)) \lor$

$(W(a) \land W(c) \land \neg W(b)) \lor$

$(W(b) \land W(c) \land \neg W(a))$

Or: $((W(a) \land W(b)) \lor (W(a) \land W(c)) \lor (W(b) \land W(c))) \land \neg (W(a) \land W(b) \land W(c))$

Or even simply $W(c)$, *assuming the first part is correct*.

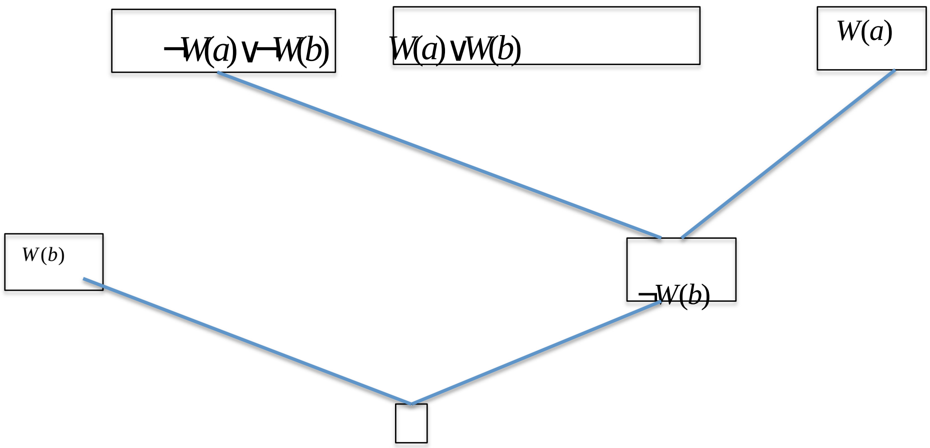
b) Both b and c should have a wumpus. Therefore we can write

$W(b) \land W(c)$

c) First convert the KB into CNF form. E.g.,

$(W(a) \lor W(b)) \land (\neg W(a) \lor W(b))$
In CNF form: $(W(a) \lor W(b)) \land (\neg W(a) \lor \neg W(b))$ etc.

To prove, add the contradiction (i.e. W(a)) of the query to the KB. Then resolve W(a) with ($\neg W(a) \lor \neg W(b)$) to conclude $\neg W(b)$.
Apply AND-elimination on W(b) $\land W(c)$ to get W(b). Resolve this with $\neg W(b)$ obtained above to get the contradiction.



Problem 5 (10 points)

This is a planning problem to be solved using GraphPlan algorithm. The initial state is represented as $\{R,H,Q\}$. The goal state is $\{D,P, ,\neg\ R\ \}$. There are four actions (Cx, Wx, Tx, and Vx) of which preconditions and effects (in terms of add and delete) are shown in the following figure.

Action	Precond	Add	Delete
Cx	{H}	{D}	{}
Wx	{Q}	{P}	{}
Tx	{}	{}	{H, R}
Vx	{}	{}	{Q, R}

a) (1 point) Under which conditions does a plan graph level off (i.e., stops expanding)?

b) (3 points) Starting from state level zero (i.e., S_0) draw the graph of the plan (i.e, the state and the action levels with mutex relations) applying the GraphPlan algorithm. Expand the graph as long as it is necessary for obtaining a plan. Use "NOP" for the persistence (no operation/maintenance) actions.

c) (3 points) Is it possible to find a plan for this problem at state level S_1? In case it is possible, write down the plan. In case it is not possible, justify your answer by writing down the mutex relation(s) that hinders the extraction of a plan from the graph. Write down each mutex relation (both those between the states and between the actions) and explain why it is mutex.

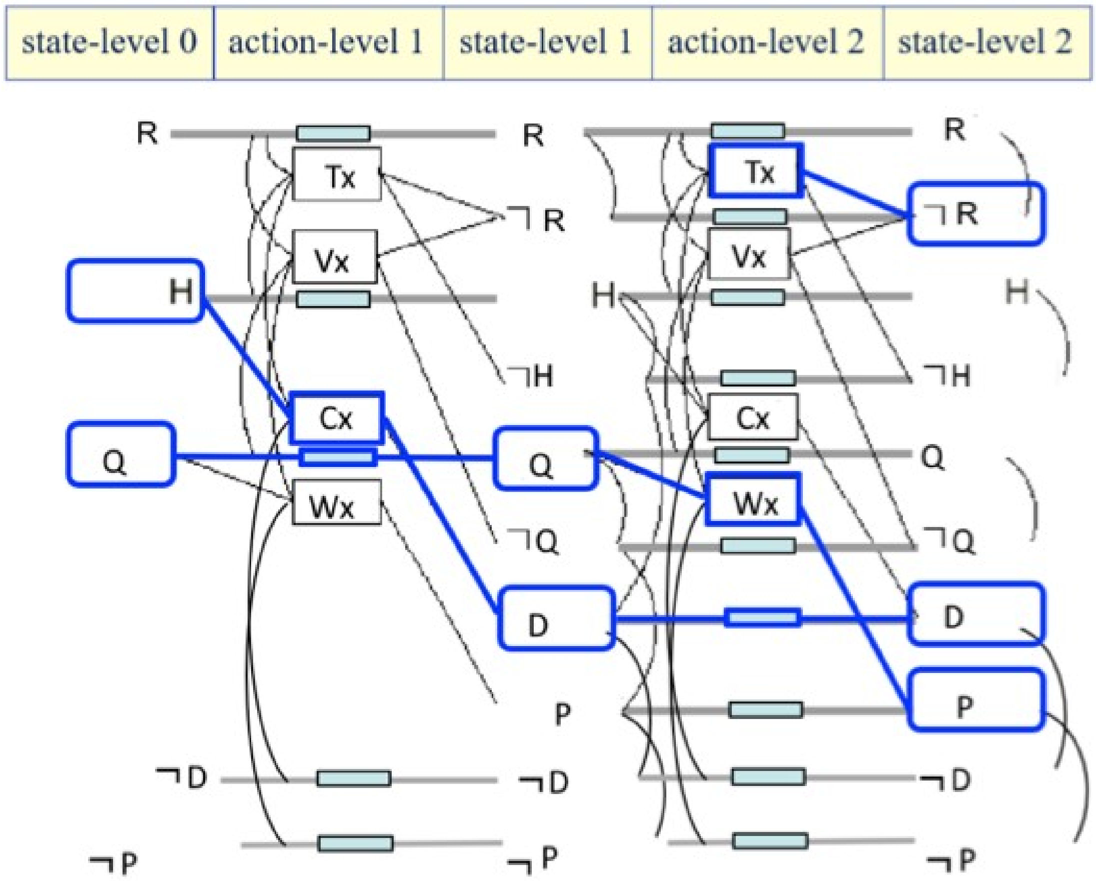
d) (3 points) Are the following plans in (1) and (2) below valid plans, according to your answers above? Explain why or why not.

1. Cx => Wx => Tx
2. Wx => Cx => Tx

ANSWER:

a) In order to leave off, the last state must include all goals in the goal state without mutex between them, and when back-searching on the graph there must be actions without mutex at each action level so that this chaining goes back to the initial state. Or the two consequtive states are equal- no change any longer.

b)



c) At S1: It is not possible to find a plan at this level. Because: mutex between Tx and Cx, and Vx and Wx.

Expand with A1 and S2. At A0 there is mutex between Cx and Tx as well as Wx and Vc. At S2 the goal $\{D, P, \neg R\}$ seems possible. Which actions enable this? $\{Cx, Wx, NO(, \neg R)\}$ is possible, without mutex.Now, need to check if the new goal $\{H, Q, , \neg R\}$ is possible at A0. No action set for this at A0. Backtract to S2, try another action set. What about $\{Cx, NO(P), Vx)\}$. This is OK. Then Try to find a nonmutex set of actions in A0 to get the newest goal $\{H, P\}$. Yes: $\{NO(H), Wx\}$. New goal now $\{H, Q\}$ which is possible at S0.Done. There are indeed more than one possible plans.

d) In our question, both plans (1) and (2) are valid plans, because $\{C_x, W_x\} \Rightarrow T_x$ is a valid plan.

Problem 6 (10 pts, 2 pts each question)
Choose the correct answers (one for each question) to the questions below.

a) What is the primary drawback of hill-climbing search?

A. The search can get stuck in a local maximum

B. The algorithm requires a lot of memory

C. The search can get stuck in a global maximum

D. The result depends strongly on the temperature schedule

ANSWER: A

b) When we illustrate the Minimax algorithm, we use the symbols 4 and 5 for nodes in the search tree. A variation of the Minimax algorithm additionally uses the symbol for some of the nodes. What does this symbol represent?

A. A second opponent in a multiplayer game

B. A game rule has been broken

C. An element of chance in the game

D. An estimated score, due to cutoff

E. A tie between the players

ANSWER: C

c) Which of the following researchers did not participate in the Dartmouth conference where the name "artificial intelligence" was coined?

A. John McCarthy

B. Marvin Minsky

C. Herbert Simon

D. Allen Newell

E. Alan Turing

ANSWER: E. Alan Turing

d) Which of the following is the main inference mechanism in semantic networks?

A. Resolution refutation

B. Generalized modus ponens

C. Inheritance

D. Skolemization

E. De Morgan's law

ANSWER: C. inheritance .

e) Which of the following would you choose as the unifier for $UNIFY\ (Loves(Per,x),Loves(y,z))$? Explain why. No points will be given without correct explanation.

A. (y/Per, x/z)

B. (y/Per, x/Per, z/Per)

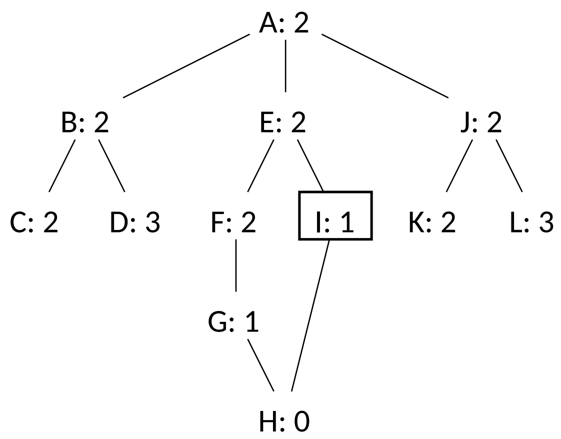
C. (y/Per, x/z, z/Siri)

D. none of the above are suitable unifiers.

ANSWER: A. (y/Per, x/z). Most general is chosen.

Problem 7 (10 points)

Consider the following search problem:



The initial node is A and the goal node is H. All step costs are 1, and a heuristic value is given for each node in the figure (for example, the heuristic value for node D is given as 3). Assume that there is a goal-test function which will be called by the search algorithm every time it needs to determine whether a node is a goal node.

a) (2 points) Using breadth-first search (BFS): What is the sequence of nodes for which the goal-test function will be called? Write the letters for each node, and in the correct order.

ANSWER: ABEJCDFIKLGH

b) (1 point) Using breadth-first search (BFS): What will be returned as the shortest path to the goal?

ANSWER: AEIH

c) (6 points) Using A*: What is the sequence of nodes for which the goal-test function will be called? Whenever several nodes have the same estimated cost, always choose the node that comes first alphabetically.

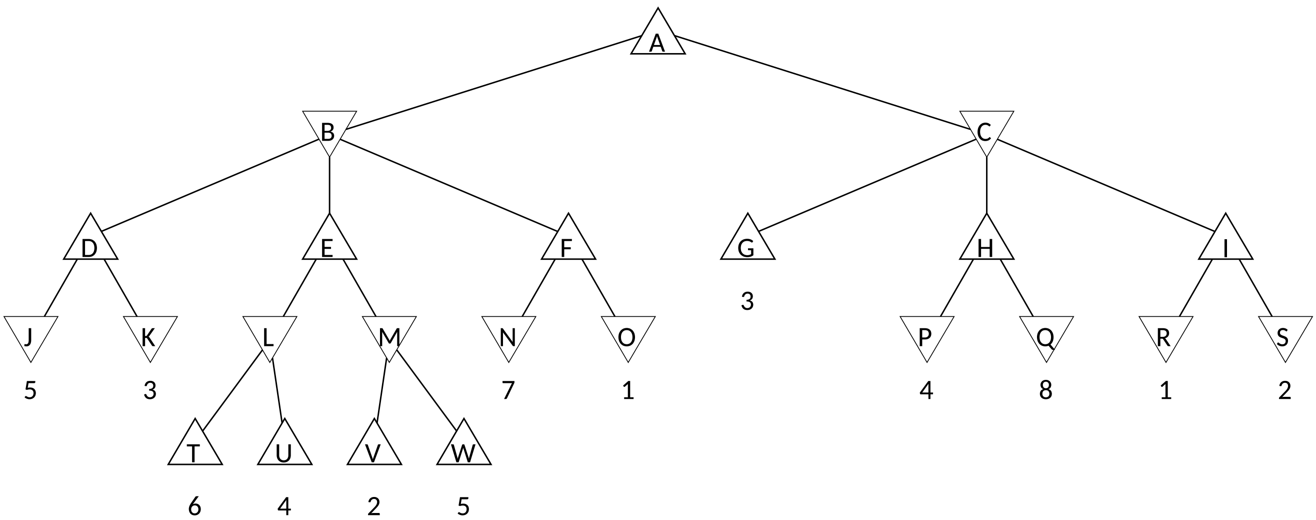
ANSWER: ABEIH

e) (1 point) Using A*: What will be returned as the shortest path to the goal?

ANSWER: AEIH

Problem 8 (10 points)

Consider the following Minimax tree:



The leaf nodes have their final utility values given as numbers below them.

a) (2 points) What is the final value in node A, after running Minimax?

ANSWER: 4

b) (3 points) What is the final value of the rest of the internal nodes, after running Minimax? Write the values for the nodes in alphabetical order, and do not include the root node or the leaf nodes. In other words, write the numbers in the order B, C, D, E, F, H, I, L, M.

ANSWER: 425478242

c) (5 points) Which nodes will be pruned when using alpha-beta pruning on this tree (assuming nodes are evaluated from left to right)? Give the answer as a list of the nodes' letters in alphabetical order, and include *all* of the nodes in the branches that are pruned.

ANSWER: HIOPQRSW

GOOD LUCK!

English. Bokmål på side 7, nynorsk på side 11. I tilfelle du er usikker på betydningen av noen av begrepene (noen er ikke så letteå oversette), se på den engelske versjonen. Answers in red text. If the student gets portions of the answer correct, reward accordingly.

Problem 1

10 points total, 1 point each subtask. Answer each question with *True* or *False*.

(a) Artificial intelligence was born when Alan Turing formulated the Turing Test. False, the field is considered to be born in 1956 with the first AI workshop at Dartmouth College.

(b) Early advances in artificial intelligence were met with skepticism and doubt. False, people were over-enthusiastic.

(c) The agent function maps percept sequences to action. True

(d) The real world is fully observable. False

(e) A taxi driving from A to B in traffic operates in a deterministic environment. False

(f) A simple reflex agent has a small but limited short-term memory. False, a simple reflex agent has no memory.

(g) Goal-based agents often rely on search and planning to find their goal. True

(h) A utility-based agent is better suited in the real world than a learning agent, since it is able to estimate its own utility. False, a learning agent is better suited to deal with the real world.

(i) Learning is necessary for complex agent behaviour to arise in a multi-agent setting. False, complex behaviour can arise from very simple agents.

(j) Learning helps in a stochastic and continuous environment. True

Problem 2

10 points total, 1 point each subtask. Answer each question with *True* or *False*.

(a) Intelligent agents are supposed to optimize their performance measure. False, they are supposed to maximize their performance measure - to optimize the performance measure does not make sense.

(b) The vacuum world is not a toy problem. False, it is a toy problem since it is an overly simplistic problem, also categorized as such in section 3.2.1.

(c) Redundant paths in a search tree are impossible to avoid. False

(d) GRAPH-SEARCH is the same as TREE-SEARCH, only with history. True

(e) Time complexity and space complexity are the two best ways to evaluate the performance of an algorithm. False, you must also consider whether it is complete and optimal in addition to the two mentioned above.

(f) Blind search is also known as heuristic search. False

(g) A graph with branching factor b and depth d can be solved in most cases by uninformed search. False, since the time/space complexity will be exponential, it cannot be solved in most cases by uninformed search. It can be solved in a few cases where b and d are small.

(h) Bidirectional search reduces the time complexity with the square root. True

(i) An heuristic estimates the cheapest cost from one node to the goal node, even if the path is impossible to execute. True

(j) A* is the best known form of best-first search. True

Problem 3

10 points in total, points indicated for each subtask. Express tasks a, b and e using first-order logic, otherwise follow the instructions.

(a) (1 point) All lectures are fun. $\forall x Lecture(x) => Fun(x)$

(b) (1 point) There exists a lecture that is not fun. $\exists x Lecture(x) \wedge \neg Fun(x)$

(c) (2 points) Siblinghood is a symmetric relationship (i.e. write how to express this relationship in first-order logic). $\forall x,y Sibling(x,y) \iff Sibling(y,x)$

(d) (2 points) Express that "everyone dislikes vegetables" in two ways, using the "FOR ALL" quantifier in one sentence and "THERE EXISTS" quantifier in the other sentence, and the same predicate in both sentences.
$\forall x \neg Likes(x, Vegetables)$ is the same as $\neg \exists x Likes(x, Vegetables)$

(e) (4 points) Some siblings have di↵erent parents.
$\exists x, y Sibling(x,y) \wedge (\exists p Parent(p,x) \wedge \neg Parent(p,y))$

Problem 4

10 points in total, 2 points each subtask.

(a) What is another word for *unification*? Substitution. See 9.2.2 in AIMA, unification is described as $UNIFY\ (p,q) = \theta$ where $SUBST(\theta,p) = SUBST(\theta,q)$

(b) What is the purpose of *Universal Instantiation*? Infer any sentence obtained by substituting a ground term for the variable, it replaces a variable with a term to form a new sentence.

(c) *Existential Instantiation* is a special case of a more general process. What is the name of this general process? Skolemization.

(d) What is the best known programming language that builds on backward chaining? Prolog

(e) What is conjunctive normal form, and what is it used for? CNF is a conjunction of clauses where each clause is a disjunction of literals. It is used for first-order resolution.

Problem 5

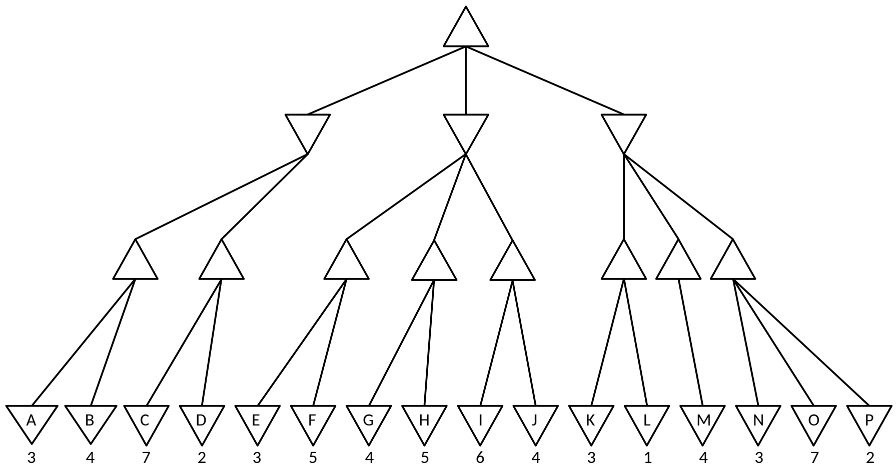
15 points. Points indicated in each subtask.

(a) (2 points) What is the name of the tree structure in the figure below? Minimax tree.

(b) (2 points) What does it represent? An adversarial game.

(c) (5 points) Write down the node values that are missing in a breadth-first manner. 5 4 5 3 4 7 5 5 6 3 4 7

(d) (6 points) Apply alpha-beta pruning and write down the leaf nodes that won't get expanded. D J M N O P



Problem 6

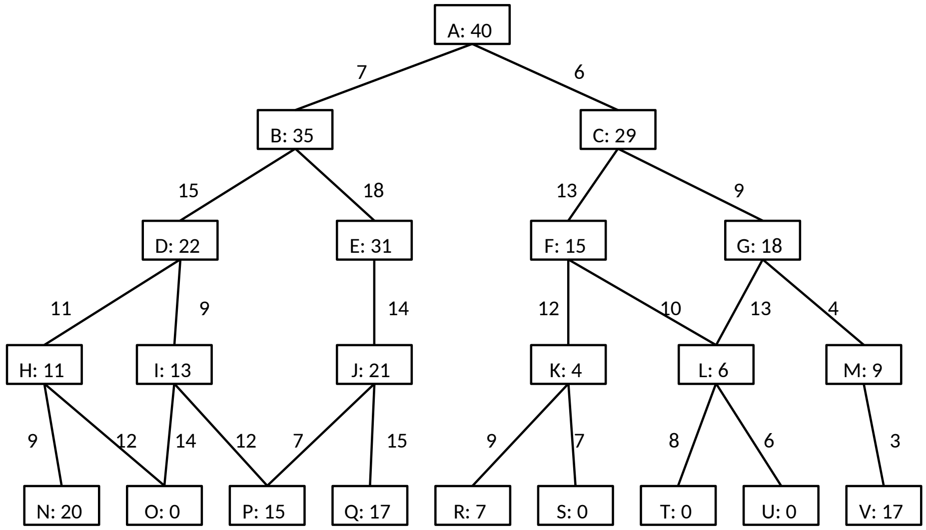
20 points. Points indicated in each subtask.

(a) (2 points) A* belongs to which class of search algorithms? Informed.

(b) (2 points) What is the worst-case time complexity of A*? Exponential, if the search space is unbounded. $O(b^d)$ It is sufficient to mention that it is exponential.

(c) (2 points) What does it mean to use an *admissible heuristic* in A* search? The heuristic must not overestimate the distance to the goal, i.e. it must be an optimistic estimate.

(d) (9 points) In the figure below, each node is labeled with the heuristic function for that node, e.g. node A has heuristic function value 40. Apply A* search to the tree and write down $f(n) = g(n)+h(n)$ for each node the algorithm visits (i.e. generates), e.g. the starting node would be written $A(40)$. Assume we visit child nodes from left to right. A(40) B(42) C(35) F(34) G(33) L(34) M(28) V(39) T(36) U(34) Note: if the student has included K(35) after V(39) this is also OK. UNIFORM-COST-SEARCH in Figure 3.14 of AIMA says that the algorithms pops a node from the frontier with the lowest cost, so this will depend on the implementation of the ordering when two nodes are of equal value.



(e) (2 points) List the nodes along the final path between the start state and the end state, using A* search. A C G L U

(f) (3 points) What is the biggest drawback of the A* algorithm? All the expanded nodes are kept in memory, so it will run out of space before it runs out of time. Therefore it does not scale well.

Problem 7

10 points. Points indicated in each subtask.

(a) (2 points) What are the best known examples of information retrieval systems? Web search engines.

(b) (3 points) What are the three essential elements in information retrieval? 1) A corpus of documents, 2) a query, 3) a result set of (ranked) relevant documents.

(c) (2 points) What separates information extraction from information retrieval? Information extraction is about acquiring knowledge from documents, whereas information retrieval is about finding the relevant documents given a query. The former builds upon the results of the latter.

(d) (3 points) What is the limiting factor in information extraction? The actual natural language processing, currently there does not exist an AI that understands text on a human level, in particular when it comes to ambiguity.

ENGLISH (Oppgaver på bokmål starter på side 8, og nynorsk på side 15)

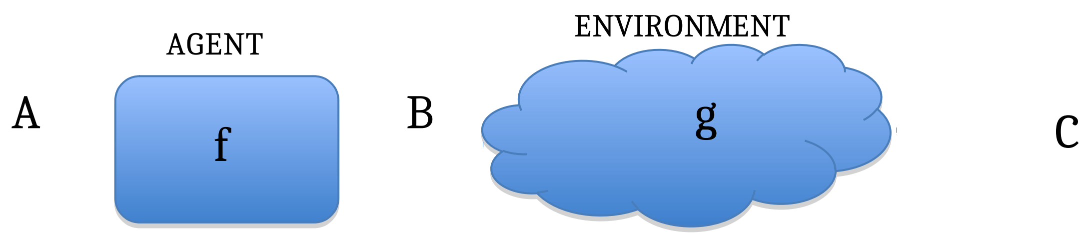
All problems have equal weight, i.e., 10 points.

Problem 1

Answer the following sentences with True or False. *A* is a special case where you should choose between *a* and *b*. For some of the sentences you are explicitly asked to explain your answer.

A. Which one of these does a rule-based agent system (like the one on a classical expert system) have in its *agent function*? a)contention scheduling algorithm b) conflict resolution algorithm? Answer with *a* or *b*. (ANSWER: b)
B. In a rule based system, when there are more than one rules that match the current situation and when the agent wants to explore a new/different (than the one in focus last time) hypothesis, the agent prefers the rule that has in its left side (of the a ▭ b) elements that are most recently updated in the working memory of the agent. Is this statement True or False? If false, describe why – very briefly. (ANSWER: F)
C. The first phase of GraphPlan can 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 present and not mutex in the last layer. Let n be the number of action layers in the graph. We can let n be the heuristic value for s. This an admissible heuristic. True or false? (ANSWER: True, because it is an underestimate of it)
D. Density is an extrinsic property of olive oil. True or false? (ANSWER: F)
E. Frame-based representation is mainly a declarative representation. True or false? (Answer: T)
F. Spreading activation and inheritance are main mechanisms of inference in rule-based systems. True or False?   (ANSWER: F)
G. Turing test measures the utility of an agent. True or false? (ANSWER: F)
H. In a rule based system, backward chaining works best when the rules have complex consequent structures. True or False? (ANSWER: F)
I. N-gram character model is a natural language model. True or False? (ANSWER: T)
J. *Sufficient* and *necessary* conditions are inference rules for sound reasoning. True or False? (ANSWER: F)

Problem 2



The above figure illustrates an agent system. The two main components (depicted as a square and a cloud shape respectively in the figure) are the agent and the environment modules.

A. In the agent terminology what is the generic/abstract names of each of A, B and C in the figure? (ANSWER: A is (perceived) current state of envir., B: action, C: envir state after execution of B.
B. What is the ***f*** and the ***g*** function called in the agent terminology? What role does each play in an agent system? (ANSWER: f: agent function, action selection fn, g: state transformer fn of the environment)
C. Is there a relationship between A and C? Explain your answer – very briefly. (ANS: A is perception of C, but to simplify, it is often assumed to be equal, i.e, as if the perception is perfect)

D. Assume now that the "AGENT" in the system is purely reactive.
   a. Draw a figure illustrating how the agent makes its decisions, i.e., illustrate what it takes into consideration and what type of decision(s) it takes?
   b. Can a purely reactive agent predict the consequences of its action in the environment? If the answers is "yes", explain how, and if it is "no" explain why not. (ANSWER: No. For prediction, a model of world is needed but purely RA does not have that model)
E. Assume now that the agent is a goal-based agent and its current task is to achieve a certain goal in the block world ENVIRONMENT . For example, ON(X ,Y); move (X ,Table) etc, .
   a. Draw a figure illustrating how the agent makes its decisions, i.e., what it takes into consideration and how it does deliberate (think)?.
   b. Can a goal-based agent predict the consequences of its action in the environment? If the answers is "yes", explain how, and if it is "no" explain why not. (ANSWER: yes ++)

Problem 3

Choose suitable predicates and translate the following sentences into first order predicate logic:

A. Cats are animals.

$$\forall x\ (Cat(x) \Rightarrow Animal(x))$$

B. Pusur is a cat.

$$Cat\ (Pusur)$$

C. Every dog owner is an animal lover.

$$\forall x\ (Exists\text{-}y\ (dog(y) \wedge Owner(x, y)) \Rightarrow Animal\text{-}lover(x)$$

D. No animal lover kills an animal.

$$\forall x\ \ \forall y\ (Animal\text{-}lover(x) \wedge Animal(y) \Rightarrow \neg Kills\ (x, y))$$

Alt: $\forall y\ \neg Exists\text{-}x\ (AnimalLover(x) \wedge Animal(y) \wedge Kills\ (x, y))$

E. Either Ole or Pusur kills Fido

$$Kills(Ole, Fido) \vee Kills\ (Pusur, Fido)$$

F. All soccer players either play with Rosenborg, or they are world-class players, or both.

∀x(Soccer-player(x) => Rosenborg-player(x) ∨ World-class-player (x))

G. All soccer players either play with Rosenborg, or they are world-class players, but not both.

∀x(Soccer-player (x) => (Rosenborg-player (x) ∨ World-class-player (x) ^ ¬ (Rosenborg-player (x) ^ World-class-player (x))

.

Problem 4

Given a set of logical sentences and a set of models for which sentences can be true or false.

A. What does it mean that a sentence is 'valid'?
Setningen er sann i alle modeller (tolkninger)

B. What does it mean that a sentence is 'satisfiable'?
Setningen er sann i minst én modell (tolkning)

C. Determine, for example by using a truth table, whether the following sentence is 'satisfiable':

$$(A \Leftrightarrow B) \wedge (\neg A \vee B)$$

Justify your answer.

A	*B*	*A ⇔ B*	*¬A ∨ B*	*(A ⇔ B) ∧ (¬A ∨ B)*
T	*T*	*T*	*T*	*T*
T	*F*	*F*	*F*	*F*
F	*T*	*F*	*T*	*F*
F	*F*	*T*	*T*	*T*

Første og siste linje viser at setningen er sann i minst én modell: => Satisfiable

D. What does it mean that an inference rule is 'sound'?
Sunn vil si sannhetsbevarende. Gitt sanne premisser er konklusjonen alltid sann.

E. Prove whether the following inference rule is sound or not:

$$\frac{P \Rightarrow Q,\ Q}{P}$$

P	***Q***	***P ⇒ Q***
T	T	T
T	F	F
F	T	T
F	F	T

Tredje linje viser at både Q og P=>Q er sann mens P er falsk. => Ikke sunn.

Problem 5

A. What is a state space? List the most important components of a state space, and describe the role of each component.

i. *Initial state, possible actions, transition model*

B. Explain the meaning of the following two concepts:

- Heuristic

- regler/kunnskap for å velge de veier i et tilstandsrom som har best sjanse for å føre til en brukbar løsning

- Heuristic search

- søkemetode som bruker kunnskap/heuristikker for å søke i tilstandsrom
- informert søk

C. Given the evaluation function for heuristic search in the form f(n) = g(n) + h(n). What do the terms mean?

g(n) er reell avstand fra startnoden, h(n) er estimert avstand til målnoden.
f(n) = g*(n) + h*(n), der h*(n) er reell kostnad fra aktuell node til tilmålnoden.*

D. Define the A* algorithm
A er algoritmen der h(n) velges lik eller mindre enn h*(n). Traveling salesperson-problemet, med h(n) lik luftlinjeavstanden til målnoden, er et godt eksempel.*

Problem 6

A. True or False?: Greedy Best-First search using the heuristic h(n) = 0 for all states n, is guaranteed to find an optimal solution.

Usant (since all nodes will have f(n)=0, result will in general depend on how ties are broken)

B. True or False?: If h1 and h2 are both admissible heuristics, it is always preferable to use the heuristic h3(n) = max(h1(n), h2(n)) over the heuristic h4(n) = min(h1(n), h2(n)).

Sant

C. True or False?: If h1 is an admissible heuristic and h2 is not an admissible heuristic, (h1 + h2)/2 must be an admissible heuristic.
Usant

D. Greedy Best-First search using h(n) = −depth(n) corresponds to which search method?

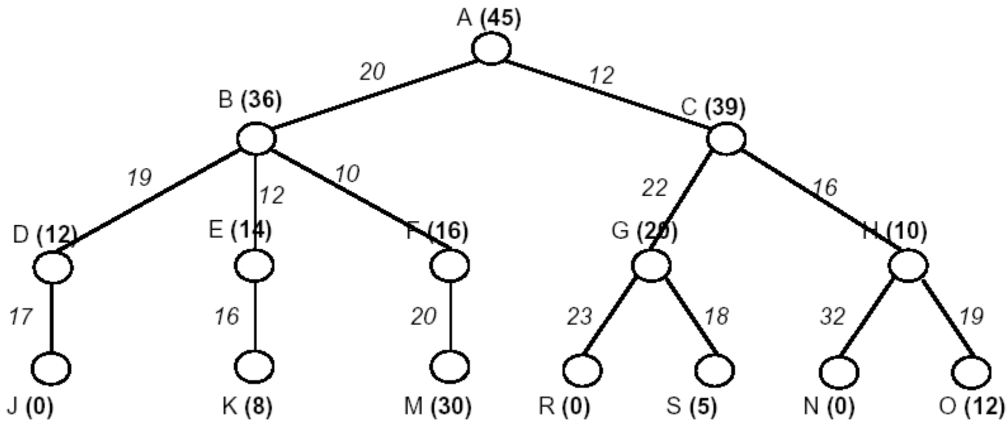
Depth-First Search

E. Say we have a search space that has a large branching factor at most nodes, there may be infinitely long paths in the search space, we have no heuristic function, and all arcs have cost 1. What search method would be good to use in this situation and why?

Depth-First Iterative Deepening because it is space efficient like DFS but also guarantees finding a solution even when there may be infinitely long paths.

Problem 7

A search tree is shown below. Node A is the initial state and the nodes J, R, og N are goal states. Each node is market (in parenthesis behind the letter) with a number corresponding to the value of the heuristic evaluation function for that node. For example: G(20).



For each of the following search strategies A and B,

- list the nodes in the order that they get expanded
- list the nodes along the final path between the initial state and the goal state:

A. Hill climbing

A(45), B(36), D(12), J(0)

-> A B D J

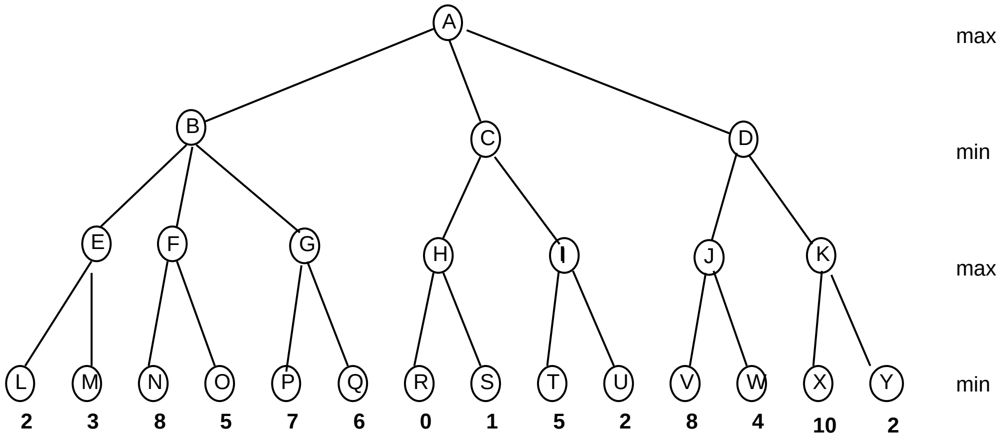
B. A* search

A(45)
C(51) B(56)
H (38) G(54) B(56)
G(54) B(56) O(59) N(60)
B(56) R(57) S(57) O(59) N(60)

E(46) F(46) D(51) R(57) S(57) O(59) N(60)
F(46) D(51) K(56) R(57) S(57) O(59) N(60)
D(51) J(56) K(56) R(57) S(57) O(59) N(60) J(56)
-> A B D J

Problem 8

Given the game tree below, in which the evaluation function values are given for the leaf nodes. Assume an alpha-beta search strategy, left to right.



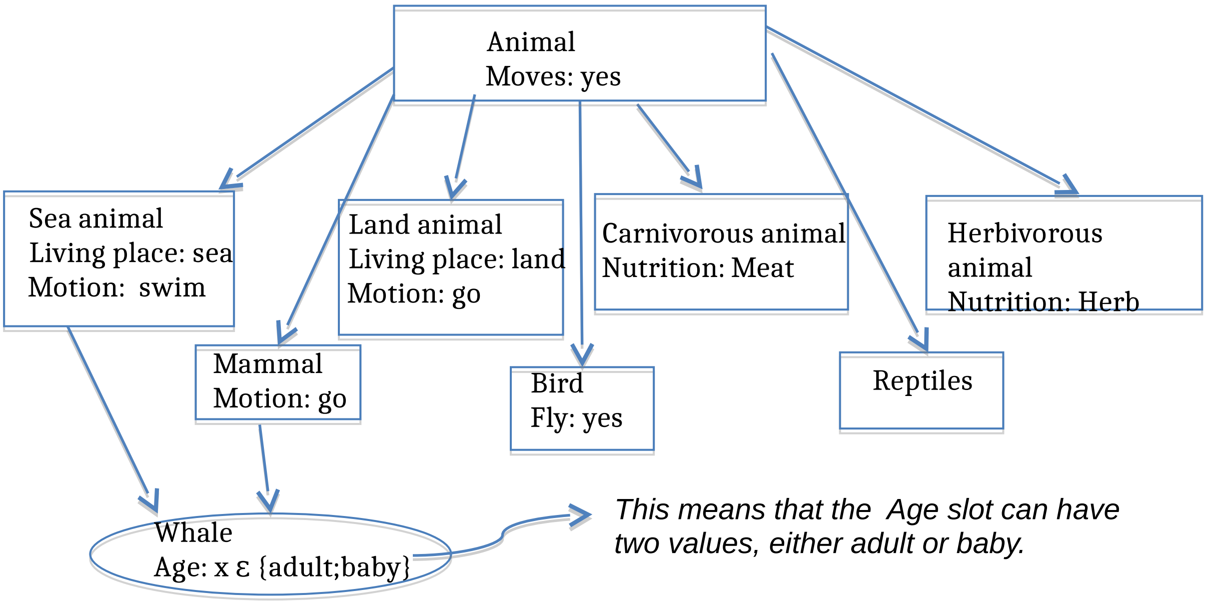
Which nodes will not be expaned? Which move will max choose in state A?

Answer: O, Q, I, T&U, Y ppruned. Max chooses D(=8)

Problem 9

A. What does a partition of a category mean? Give an example. (ANSWER: disjoint, exhaustive)
B. Does the **animal** category in the figure below have a partition? If so, write it down; write all partitions of **animal** if there are more than one. .(ANSWER: sea animal and land animal is a partition, carniv, herbiv is another one. Maybe mammal, bird, reptile is another one. It is not necessary to find absolutely all partition. What is important is to show that the student knows the difference betwen a subcategory/subclass and a partition)
C. We want the system to be able to answer the question “ How many kg does a particular whale (i.e., my-whale) eat each day?”. How would you update the knowledge base of the agent in order for it to give the correct answer to such a question? In your solution, the system shall have the knowledge that baby whales eat 5 kg food each day while adult whales eat 20 kg. A constraint is that the new knowledge base shall NOT have two additional frames for “baby whale” and “adult

whale". ". (ANSWER: add a new, instance frame myWhale under whale. Its age: baby. Another whale instance can be yourWhale, where age slot has value: adult. Then the Whale frame must have an Eats slot of which value is a procedural attachment. Stg like this: If age=baby then eats= 5 kl else eats=20 kg.. Bottomline: to show that a slot value can be procedural attachment/function/code.



Problem 10

This problem is about planning and consists of two independent questions (A and B).

A. Suppose you are implementing a planning mechanism for the block-world applications. **Result** is a function which, given action ***a*** and environmental state ***s***, computes the the state of the environment after execution of ***a***. Assume you have the following two alternatives to define **Result**:

Result1 (s, a) = ([s ∪ [Add(a)] - Del(a))
Result2 (s, a) = ([s ∪ Del(a)]- Add(a))

where ***Del*** and ***Add*** correspond to what becomes True and False, respectively, as the result of action ***a***.

Would a choice between adopting Result1 or Result2 make a difference on the planning result? Choose one of these 3 options: a) No difference between Result1 and Result2, b) Result1 is the correct function, c) Result2 is the correct function.

Motivate your answer by considering the following PDDL for a blocks-world problem:

Action MoveToTable(b, x)

PRECOND: On(b, x) ^ Clear(b) ^ Block(b) ^ (b ≠x)
EFFECT: On(b,Table) ^ Clear(x) ^ ¬ On(b, x)

Notice that in the PDDL snippet above, **Add** and **Del** are not separated but are put together as **EFFECT**.
Explain your answer briefly.

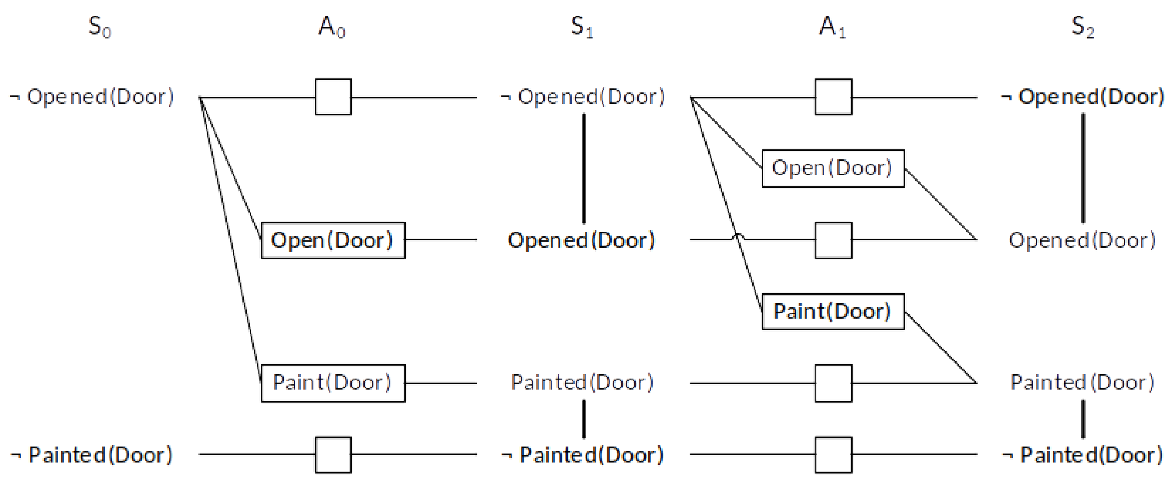
(ANSWER

The behavior does change, because the ordering of the addition and removal can be important when the same fluents are in both lists. By ordering addition before deletion, we can end up with a fluent not in the effects list, which should be. In this case MoveT oT able(B, Table) fails because On(b, Table) is added then deleted – so we are left without that condition. See the book. This case is explained there.

B. For the following planning problem

Init(¬Opened(Door) ^¬Painted(Door))
Goal(Opened(Door) ^ Painted(Door))
Action(Open(Door)
Precond: ¬Opened(Door)
Effect: Opened(Door))
Action(Paint(Door)
Precond: ¬Opened(Door)
Effect: Painted(Door)

we have the following planning graph where some mutex links are missing:



Find the missing mutex links between state literals and between actions on all levels.

Write down the list of missing mutex links for each level (both state and action levels). For each mutex link, provide an explanation of why they are mutex. Your answer will look like, for example:

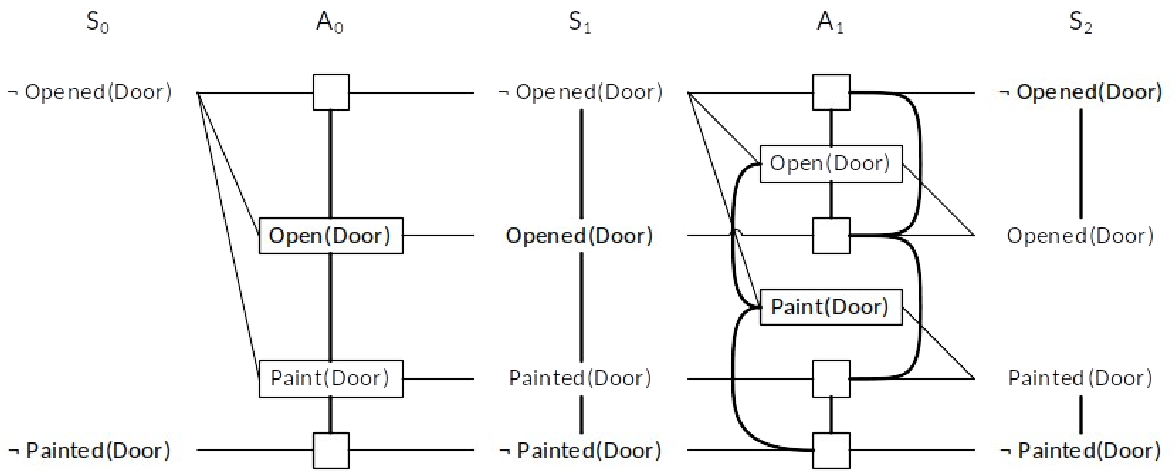
S1:

¬Opened(Door), Opened(Door) – one negates the other
¬Painted(Door), Painted(Door) – one negates the other

S2:

¬Opened(Door), Opened(Door) – one negates the other
¬Painted(Door), Painted(Door) – one negates the other

You do not need to draw the graph, just list the pairs of the state literals and the actions that have mutex relations between them, with an explanation for each pair as shown in the example above.



ANSWER: For actions, there can be several valid explanations of mutex links based on either inconsistent effect, interference or competing needs (see chapter 10.3 in the textbook or Recitation 6: Graphplan presentation on itslearning). Mutex links propagate recursively, i.e. you will need to look at mutex links in the previous level to determine mutex links for the current level.

A_0:
¬Opened(Door), Open(Door) – effect ¬Opened(Door) is inconsistent with effect Opened(Door) in S_1

Open(Door), Paint(Door) – effect Opened(Door) in S_1 of Open(Door) interferes with precondition ¬Opened(Door) of action Paint(Door) in S_0

Paint(Door), ¬Painted(Door) – effect Painted(Door) is inconsistent with effect ¬Painted(Door) in S_1

S_a:

Opened(Door), Painted(Door) – the only way to achieve them is through mutex actions Open(Door) and Paint(Door) in A_0

A_1:
¬Opened(Door), Open(Door) – effect ¬Opened(Door) is inconsistent with effect Opened(Door) in S_2

Opened(Door), Open(Door) – preconditions Opened(Door) and ¬Opened(Door) are mutex in S_1

Painted(Door), ¬Painted(Door) – effect Painted(Door) is inconsistent with effect ¬Painted(Door) in S_2

Paint(Door), ¬Painted(Door) – effect Painted(Door) is inconsistent with effect ¬Painted(Door) in S_2

¬Opened(Door), Opened(Door) – effect ¬Opened(Door) is inconsistent with effect Opened(Door) in S_2

Open(Door), Paint(Door) – effect Opened(Door) in S_2 of action Open(Door) interferes with precondition ¬Opened(Door) in S_1 of action Paint(Door)

Opened(Door), Painted(Door) - Opened(Door) and Painted(Door) in S_1 are mutually exclusive

END OF QUESTIONS
GOOD LUCK!

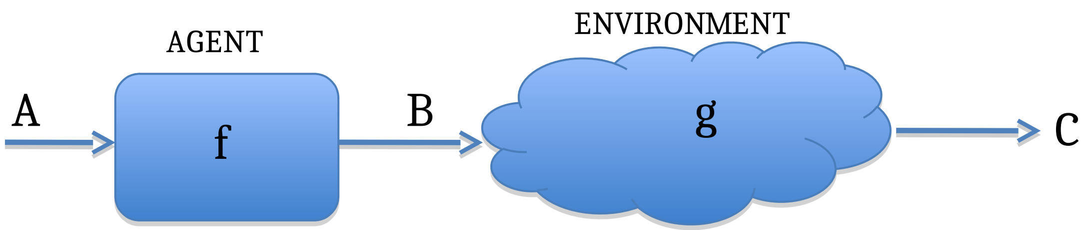
BOKMÅL
Alle oppgavene har likt vekt, 10 poeng.

Oppgave 1

Besvar følgende setninger med True eller False. A er et spesielt tilfelle hvor du må velge mellom alternativene *a* og *b*.
I noen av de setningene/spørsmålene blir du eksplisitt spurt om kort å begrunne svaret ditt.

A. Hvilken av disse har et regel-basert agent system (som i klassiske ekspert systemer) i sin ”agent function”? a) ”contention secheduling” algoritme, b) ”conflict resolution” algoritme. Svar med enten *a* eller *b*.
B. I et regel-basert system, når det er mer enn én regel som matcher den nåværende situasjonen og agenten ønsker å utforske en ny/forskjellig hypotese (enn den som var i fokus siste gangen), foretrekker agenten den regelen som i sin venstre side (av a □ b) har elementer som er oppdatert i ”working memory” nyligst.
Er denne setningen True eller False? Hvis False, beskriv kort hvorfor.
C. Den første fasen i GrafPlan kan brukes som heuristisk funksjon for ”forward”/fremover? Søk på følgende måte: Gitt en tilstand s og mål g, kjør graph-construction fasen av en GraphPlan til alle komponenter er presentert og det ikke er noe mutex på siste level. La *n* være antall action lag i grafen. Vi kan la *n* bli en heuristisk verdi for *s*. Denne er en ‘admissible’ heuristikk. True eller False?
D. Densitet er en ‘extrinsic’ egenskap av oliv oil. True eller False?
E. Ramme-basert representasjon er i utgangspunktet en deklarativ representasjon. True eller False?
F. ‘Spreading activation’ og arv er hovedmekanismer av inferens. True eller false?
G. Turing test måler ‘utility’ av en agent. True or False?
H. I regel-baserte systemer virker ‘bakward chaining ‘ (bakoverlenking) best når reglene har komplekse ”consequent” strukturer. True eller False?
I. ”N-gram character model” er en naturlig språk model. True or False?
J. Betingelsene *tilstrekkelig* (sufficient)og *nødvendig* (necessary) er inferensregler for sound (sound) resonnering. True eller False?

Oppgave 2



Figuren ovenfor illustrerer et agentsystem. De to hovedkomponentene (vist som firkant- og sky-form i figuren) er agenten (AGENT) og omgivelsen (ENVIRONMENT).

A. I agent-terminologi, hva er de generisk/abstrakt navn til hver av A, B og C i figuren?
B. Hva kalles ***f*** og ***g*** funksjonen i agent-terminologi? Hvilken rolle spiller hver av dem i et agentsystem?
C. Er det en relasjon mellom A og C? Forklar ditt svar –kort.
D. Anta nå at ”AGENT” i systemet er rent reaktiv.
    a. Tegn en figur som illustrerer hvordan agenten tar sine beslutninger, i.e., illustrer hva den tar hensyn til og hvilke beslutninger den tar.
    b. Kan rent reaktive agenter predikere konsekvensene av sine handlinger i omgivelsene? Hvis svaret er ’ja’, hvordan? Hvis svaret er ’nei’, hvorfor ikke?

E. Anta nå at agenten er en mål-basert agent og dens nåværende oppgave er å oppnå et bestemt mål i blokk-verden ENVIRONMENT. For eksempel, ON(X ,Y); move (X ,Table), osv.
    a. Tegn en figur som illustrerer hvordan en agent tar sine beslutninger, dvs hva den tar hensyn til og hvordan den funderer/tenker for å ta en beslutning?
    b. Kan mål-baserte agenter predikere konsekvensene av sine handlinger i omgivelsene? Hvis svaret er ’ja’, hvordan? Hvis svaret er ’nei’, forklar hvorfor ikke?

Oppgave 3

Velg passende predikater og oversett følgende setninger til førsteordens predikatlogikk:

A. Cats are animals.
B. Pusur is a cat.
C. Every dog owner is an animal lover.
D. No animal lover kills an animal.
E. Either Ole or Pusur kills Fido
F. All soccer players either play with Rosenborg, or they are world-class players, or both.
G. All soccer players either play with Rosenborg, or they are world-class players, but not both.

Oppgave 4

Gitt et sett av logiske setninger og et sett av modeller der setninger kan være sanne eller usanne.

A. Hva vil det si at en setning er ‘valid’?
B. Hva vil det si at en setning er ‘satisfiable’?

C. Bestem, f.eks. ved hjelp av en sannhetstabell, om følgende setning er ’satisfiable’:

$$(A \Leftrightarrow B) \wedge (\neg A \vee B)$$

Begrunn svaret.

D. Hva vil det si at en inferensregel er ‘sound’ (sunn)?
E. Bevis hvorvidt følgende inferensregel er sunn:

$$\frac{P \Rightarrow Q,\ Q}{P}$$

Oppgave 5

A. Hva er et tilstandsrom (state space)? List opp de viktige komponenter som et tilstandsrom består av, og beskriv hvilken rolle hver av komponentene har.

B. Gi en kort forklaring på følgende to begreper:
- Heuristikk
- Heuristisk søk

C. Gitt evalueringsfunksjonen for heuristisk søk på formen f(n) = g(n) + h(n). Hva betyr de enkelte leddene i funksjonen?

D. Gitt evalueringsfunksjonen i deloppgave C, hva er kriteriet for at algoritmen A* finner en optimal løsning?

E. Hvis en heuristikk er ‘admissible’, er den da også ‘consistent’? Begrunn svaret kort.

Oppgave 6

A. True eller False?: ‘Greedy Best-First’-søk som benytter heuristikken h(n) = 0 for alle tilstander n, er garantert å finne en optimal løsning.

B. True eller False?: Hvis h1 and h2 begge er ‘admissible’ heuristikker så er det alltid fordelaktig å bruke heuristikken h3(n) = max(h1(n), h2(n)) istedenfor heuristikken h4(n) = min(h1(n), h2(n)).

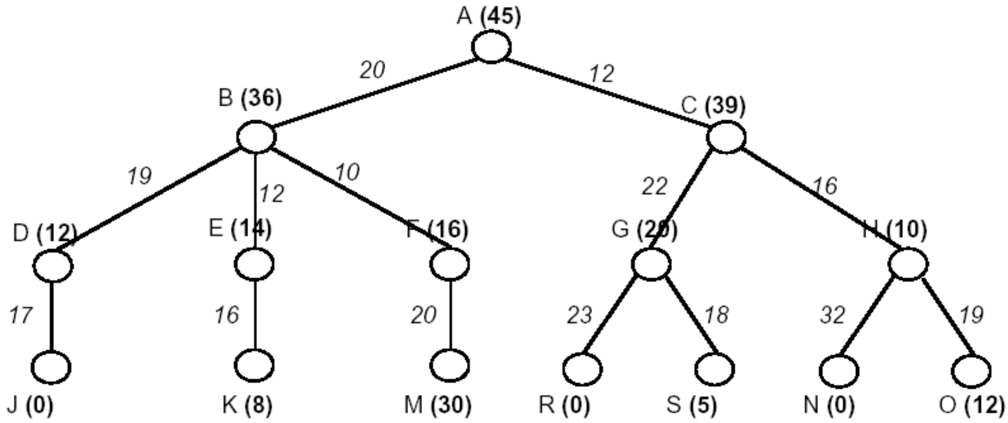
C. True eller False?: Hvis h1 er en ‘admissible heuristikk og h2 ikke er en ‘admissible’ heuristic, vil (h1 + h2)/2 være en ‘admissible’ heuristikk.

D. ‘Greedy Best-First’ -søk der h(n) = −depth(n) tilsvarer hvilken kjent søkemetode?

E. Anta at vi har et søkerom med en høy forgreningsfaktor for de fleste noder. Anta videre at vi kan ha uendlig lange veier i søkerommet, at vi ikke har noen heuristisk funksjon, og at alle forbindelser mellom to noder har kostnad 1.
Hvilken søkemetode vil det være bra å bruke i dette tilfellet, og hvorfor?

Oppgave 7

Figuren under viser et søketre. Node A er starttilstanden og nodene J, R, og N er måltilstander. Hver node er merket (i parentes etter bokstaven) med verdien av den heuristiske evalueringsfunksjonen for noden. For eksempel: G (20).



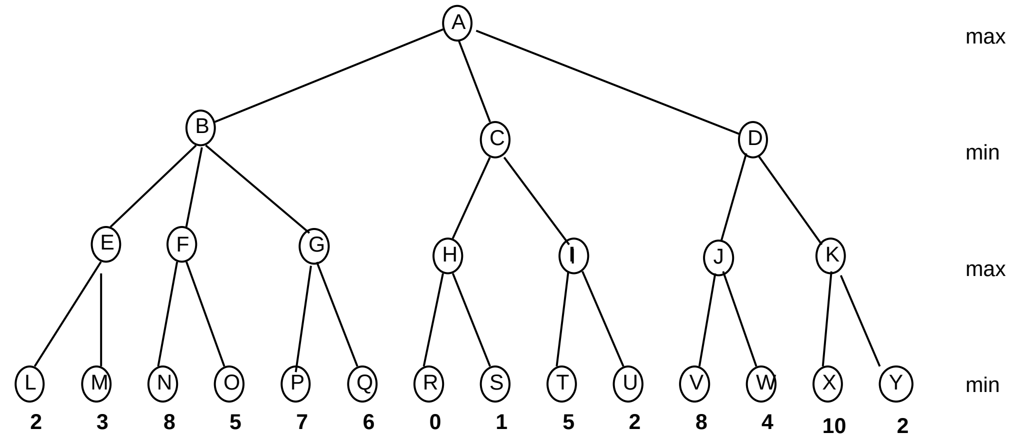
For hver av de følgende søkestrategiene A og B,

- list nodene i den rekkefølge de blir undersøkt (ekspandert)
- list nodene langs den endelige veien mellom start- og måltilstanden:

A. Bakkeklatring (hill climbing)
B. A* søk

Oppgave 8

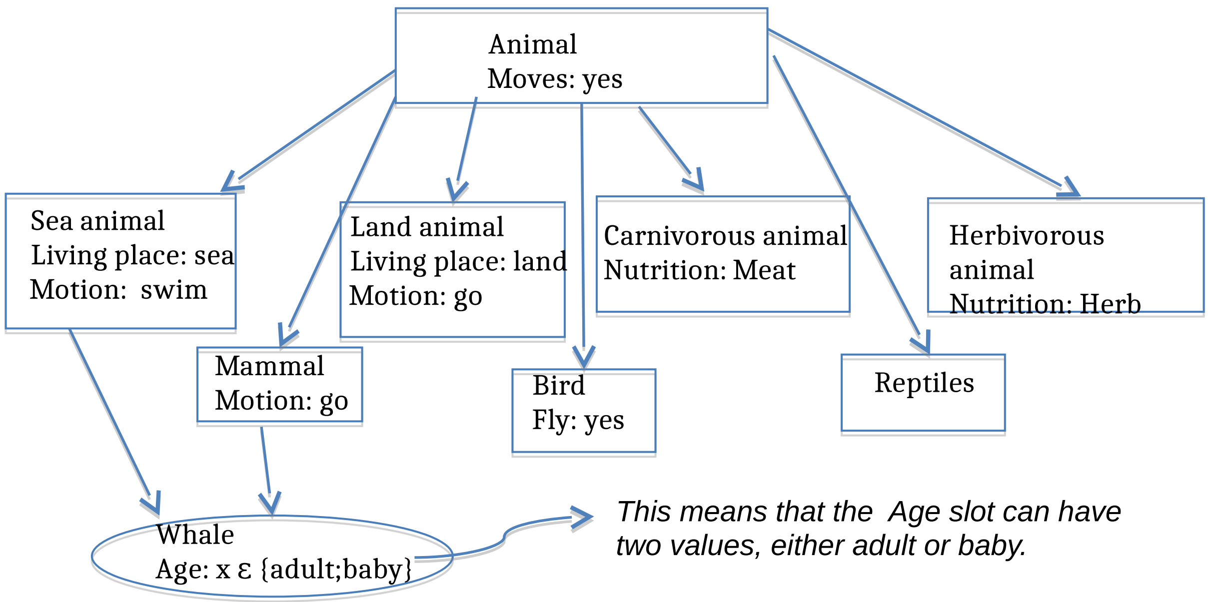
Gitt følgende spilltre, der verdier for evalueringsfunksjonen er angitt for bladnodene. Anta et alfa-beta søk, fra venstre mot høyre.



Hvilke noder trenger ikke undersøkes? Hvilket trekk velger max i tilstand A?

Oppgave 9

A. Hva betyr ”partition” av en kategori? Gi et eksempel.
B. Har kategorien **animal** i figuren nedenfor en ”partition”? I så fall skriv den ned; skriv ned alle ”partitions” av **animal** hvis det er mer en enn.
C. Vi vil at systemet skal kunne svare på spørsmålet ”Hvor mange kg mat spiser en bestemt hval (f. eks. my-whale) hver dag?”. Hvordan vil du oppdatere agentens kunnskapsbase for å få det riktige svaret på et slik spørsmål? Systemet vet at baby-hvaler spiser 5 kg mat hver dag mens voksne hvaler spiser 20 kg. En betingelse er at den nye databasen skal IKKE ha to nye rammer for “baby whale” og “adult whale”.



Oppgave 10

Denne oppgaven er om planlegging og består av to uavhengige spørsmål (A og B).

A. Anta at du utvikler en planleggingsmekanisme for block-world applikasjoner. **Result** er en funksjon som, gitt handling **a** og miljø-tilstand ***s***, beregner tilstand av miljøet etter eksekvering av **a**. Anta at du har følgende to alternativer for å definere **Result**:

Result1 (s, a) = (s ∪ [Add(a) - Del(a)])
Result2 (s, a) = (s ∪ [Del(a) - Add(a)])

hvor **Del** og **Add** tilsvarer hva som blir True eller False som resultat av handling a.

Vil valget mellom **Result1** og **Result2** gjøre noe forskjell på resultatet av planleggingen? Velg en av disse 3 mulighetene: a) Det er ikke forskjell mellom Result1 og Result2, b) Result1 er den riktige funksjonen, c) Result2 er den riktige funksjonen.

Forklar ditt svar ved å referere til følgende PDDL i forbindelse med blokk-world problemet.

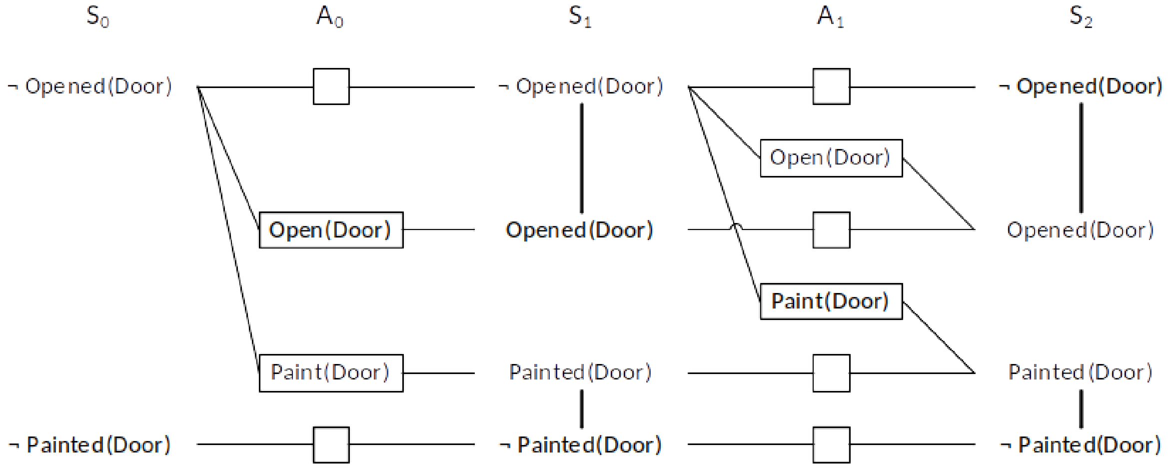
Action (MoveToTable(b, x)
PRECOND: On(b, x) ^ Clear(b) ^ Block(b) ^ (b ≠x)
EFFECT: On(b,Table) ^ Clear(x) ^ ¬ On(b, x))

Merk at i PDDL-"snippet" ovenfor så er **Add** og **Del** ikke delt opp, men puttet sammen som **EFFECT**. Forklar kort ditt svar.

B. For følgende planleggingsproblem:

Init(¬Opened(Door) ^ ¬Painted(Door))
Goal(Opened(Door) ^ Painted(Door))
Action(Open(Door)
  Precond: ¬Opened(Door)
  Effect: Opened(Door))
Action(Paint(Door)
  Precond: ¬Opened(Door)
  Effect: Painted(Door))

har vi den følgende "planning graph" hvor noen mutex-lenker mangler:



Finn de manglende mutex lenker mellom tilstands-literals og handlings-literals på alle nivå. Skriv ned listen av manglende mutex-lenker for hvert nivå (både tilstands- og handlings- nivåer). For hver mutex-lenke, gi en forklaring av hvofor de er mutex. Svar ditt vil, for eksempel, slik ut:

S1:
¬Opened(Door), Opened(Door) – one negates the other
¬Painted(Door), Painted(Door) – one negates the other

S2:
¬Opened(Door), Opened(Door) – one negates the other
¬Painted(Door), Painted(Door) – one negates the other

Du trenger ikke tegne grafen, bare liste de parene av tilstands- og handlings-literaler som har mutex-relasjoner mellom seg, sammen med en forklaring for hvert par som vist i eksempelet ovenfor.

SLUTT på SPØRSMÅL
LYKKE TIL!

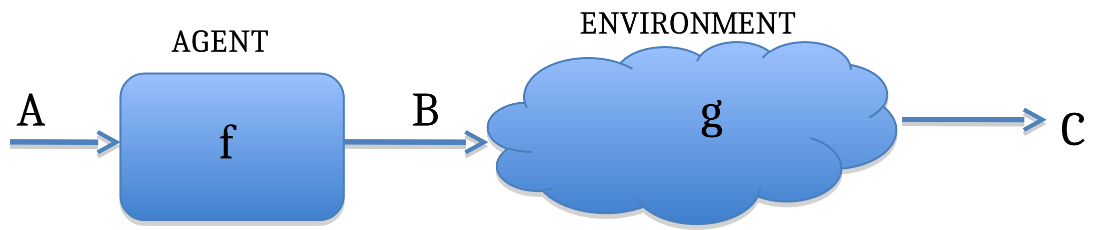
NYNORSK
Alle oppgåvene har likt vekt, 10 poeng.

Oppgåve 1

Besvar følgjande setningar med True eller False. A er eit spesielt tilfelle der du må velje mellom alternativa a og b.
I nokre av dei setningane/spurnadene blir du eksplisitt spurt om kort å begrunne svaret ditt.

A. Kva for ein av desse har eit regel-basert agent system (som i klassiske eksperten system) i agenten ”sin function”? a) ”contention secheduling” algoritme, b) ”conflict resolution” algoritme. Svar med anten a eller b.
B. I eit regel-basert system, når det er meir enn éin regel som matchar den noverande situasjonen og agenten ynskjer å utforske ein ny/ulik hypotese (enn han som var i fokus siste gongen), foretrekker agenten den regelen som i den venstre sida si (av a->b) har element som er oppdatert i ”working memory” nylegast.
Er denne setninga True eller False? Viss False, skildr kort kvifor.
C. Den første fasen i GrafPlan kan brukast som heuristisk funksjon for ”forward”/framover? Søk på følgjande måte: Gjeve ein tilstand s og mål g, køyr graph-construction fasen av ein GraphPlan til alle komponentar er presentert og det ikkje er noko mutex på siste level. La n vere mengder action lag i grafen. Vi kan la n bli ein heuristisk verd for s. Denne er ein ‘admissible’ heuristikk. True eller False?
D. Densitet er ein ‘extrinsic’ eigenskap av oliv oil. True eller False?
E. Ramme-basert representasjon er i utgangspunktet ein deklarativ representasjon. True eller False?
F. ‘Spreading activation’ og arv er hovudmekanismar av inferens. True eller false?
G. Turing test måler ‘utility’ av ein agent. True or False?
H. I regel-baserte system verkar ‘bakward chaining ‘ (bakoverlenking) best når reglane har komplekse ”consequent” strukturar. True eller False?
I. ”N-gram character model” er ein naturleg språk modell. True or False?
J. Betingelsene *tilstrekkeleg* (sufficient)og *naudsynt* (necessary) er inferensregler for sunn (sound) resonnering. True eller False?

Oppgåve 2



Figuren ovanfor illustrerer eit agentsystem. Dei to hovudkomponentane (vist som firkant- og sky-form i figuren) er agenten (AGENT) og miljøet (ENVIRONMENT).

A. I agent-terminologi, kva er dei generisk/abstrakt namn til kvar av A, B og C i figuren?
B. Kva blir funksjonene **f** og **g** kalla i agent-terminologi? Kva for rolle spelar kvar av dei i eit agentsystem?
C. Er det ein relasjon mellom A og C? Forklar svaret ditt –kort.
D. Anta no at "AGENT" i systemet er reint reaktiv.
   a. Teikn ein figur som illustrerer korleis agenten tek sine avgjerder, i.e., illustrer kva han tek omsyn til og kva for avgjerder han tek.
   b. Kan reint reaktive agentar predikere konsekvensane av handlingane sine i omgivnadene? Viss svaret er 'ja', korleis? Viss svaret er 'nei', kvifor ikkje?

E. Anta no at agenten er ein mål-basert agent og den noverande oppgåva dens er å oppnå eit bestemt mål i blokk-verda ENVIRONMENT. For eksempel, ON(X ,Y); move (X ,Table) osb.
   a. Teikn ein figur som illustrerer korleis ein agent tek sine avgjerder, dvs. kva han tek omsyn til og korleis han funderer/tenkjar for å ta ein avgjerd?
   b. Kan mål-baserte agentar predikere konsekvensane av handlingane sine i omgivnadene? Viss svaret er 'ja', korleis? Viss svaret er 'nei', forklar kvifor ikkje?

Oppgåve 3

Vel passande predikat og omset følgjande setningar til første ordens predikatlogikk:

A. Cats are animals.
B. Pusur is a cat.
C. Every dog owner is an animal lover.
D. No animal lover kills an animal.
E. Either Ole or Pusur kills Fido
F. All soccer players either play with Rosenborg, or they are world-class players, or both.
G. All soccer players either play with Rosenborg, or they are world-class players, but not both.

Oppgåve 4

Gjeve eit sett av logiske setningar og eit sett av modellar der setningar kan vere sanne eller usanne.

A. Kva vil det seie at ei setning er 'valid'?
B. Kva vil det seie at ei setning er 'satisfiable'?

C. Avgjer, t.d. ved hjelp av ein sannhetstabell, om følgjande setning er 'satisfiable':

$$(A \Leftrightarrow B) \wedge (\neg A \vee B)$$

Grunngjev svaret.

D. Kva vil det seie at ein inferensregel er sunn (sound)?
E. Bevis i kva grad følgjande inferensregel er sunn:

$$\frac{P \Rightarrow Q,\ Q}{P}$$

Oppgåve 5

A. Kva er eit tilstandsrom (state space)? List opp dei viktige komponentane som eit tilstandsrom består av, og skildr kva for ei rolle kvar av komponentane har.

B. Gje ei kort forklaring på følgjande to omgrep:
- Heuristikk
- Heuristisk søk

C. Gjeve evalueringsfunksjonen for heuristisk søk på forma f(n) = g(n) + h(n).
Kva tyder dei einskilde ledda i funksjonen?

D. Gjeve evalueringsfunksjonen i deloppgåve C, kva er kriteriet for at A* finn ei optimal løysing?

E. Viss ein heuristikk er ‘admissible’, er han då òg ‘consistent’? Grunngjev kort svaret.

Oppgåve 6

A. True eller False?: ‘Greedy Best-First’ -søk som nyttar heuristikken h(n) = 0 for alle tilstandar n, er garantert å finne ein optimal løysing.

B. True eller False?: Viss h1 and h2 begge er ‘admissible’ heuristikker så er det alltid fordelaktig å bruke heuristikken h3(n) = max(h1(n), h2(n)) i staden for heuristikken h4(n) = min(h1(n), h2(n)).

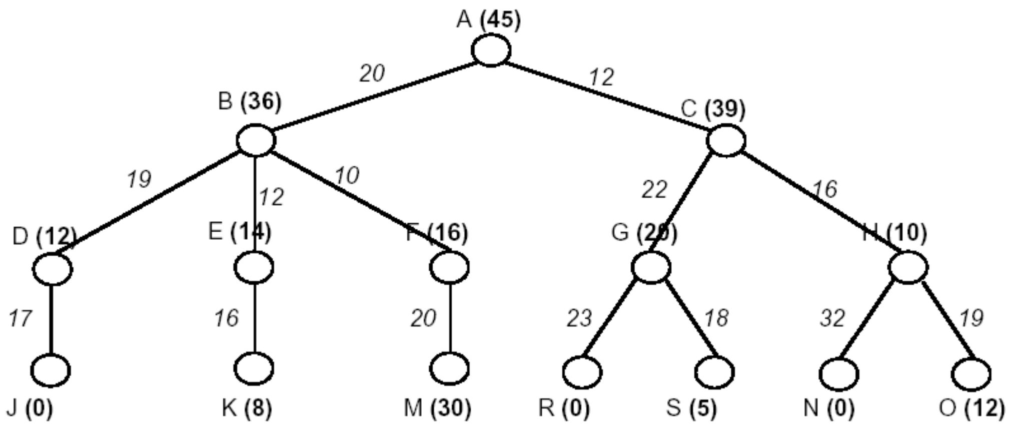
C. True eller False?: Viss h1 er ein ‘admissible heuristikk og h2 ikkje er ein ‘admissible’ heuristic, vil (h1 + h2)/2 vere ein ‘admissible’ heuristikk.

D. ‘Greedy Best-First’ søk der h(n) = −depth(n) tilsvarar kva for ein kjent søkemetode?

E. Anta at vi har eit søkerom med ein høg forgreiningsfaktor for dei fleste noder. Anta vidare at vi kan ha uendeleg lange vegar i søkerommet, at vi ikkje har nokon heuristisk funksjon, og at alle samband mellom to noder har kostnad 1.
Kva for ein søkemetode vil det vere bra å bruke i dette tilfellet, og kvifor?

Oppgåve 7

Figuren under viser eit søketre. Node A er starttilstanden og nodene J, R, og N er måltilstander. Kvar node er merkt (i parentes etter bokstaven) med verdien av den heuristiske evalueringsfunksjonen for noden. Til dømes: G (20).

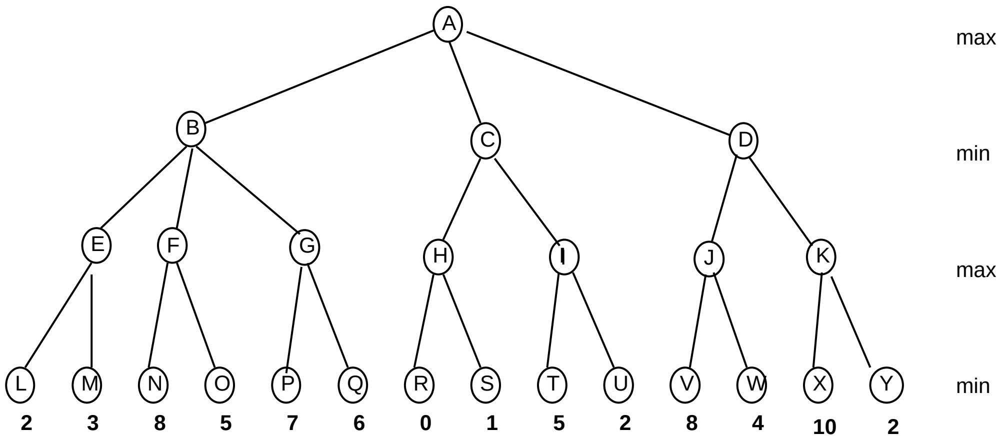


For kvar av dei følgjande søkestrategiene A og B,
- list nodene i den rekkefølgja dei blir undersøkt (ekspandert)
- list nodene langs den endelege vegen mellom start- og måltilstanden:

A. Bakkeklatring (hill climbing)
B. A* søk

Oppgåve 8

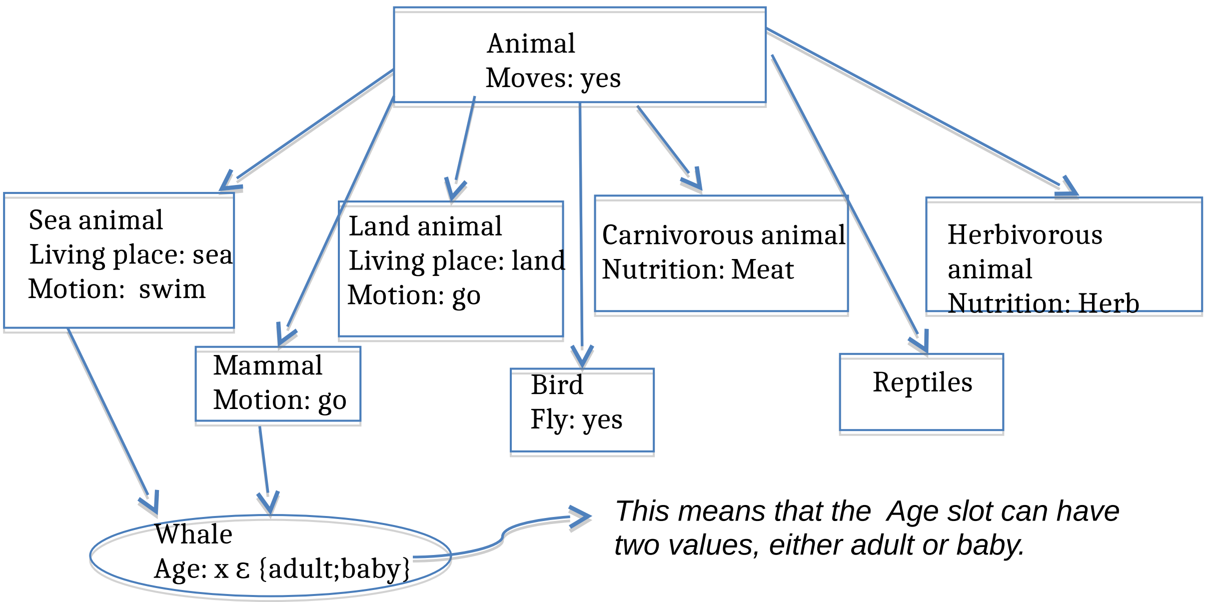
Gjeve følgjande spilltre, der verd for evalueringsfunksjonen er angjeve for bladnodene. Anta eit alfa-beta søk, frå venstre mot høgre.



Kva for noder treng ikkje undersøkjast? Kva for eit trekk vel max i tilstand A?

Oppgåve 9

A. Kva tyder ”partition” av ein kategori? Gje eit døme.
B. Har kategorien animal i figuren nedanfor ein ”partition”? I så fall skriv han ned; skriv ned alle ”partitions” av animal viss det er meir ein enn.
C. Vi vil at systemet skal kunne svare på spurnaden ”Kor mange kg mat et ein bestemd hval (f. eks. my-whale) kvar dag?”. Korleis vil du oppdatere agenten sin kunnskapsbase for å få det riktige svaret på eit slik spurnad? Systemet veit at baby-hvaler et 5 kg mat kvar dag medan vaksne hvaler et 20 kg. Ein betingelse er at den nye databasen skal IKKJE ha to nye ramar for “baby whale” og “adult whale”.



Oppgåve 10

Denne oppgåva er om planlegging og består av to uavhengige spørsmål (A og B).

A. Anta at du utviklar ein planleggingsmekanisme for block-world applikasjonar. **Result** er ein funksjon som, gjeve handling **a** og miljø-tilstand ***s***, bereknar tilstand av miljøet etter eksekvering av a. Anta at du har følgjande to alternativ for å definere **Result**:

Result1 (s, a) = (s ∪ [Add(a) - Del(a)])
Result2 (s, a) = (s ∪ [Del(a) - Add(a)])

kor **Del** og **Add** tilsvarar kva som blir True eller False som resultat av handling a.

Vil valet mellom **Result1** og **Result2** gjere noko skilnad på resultatet av planlegginga? Vel ein av desse 3 høva: a) Det er ikkje skilnad mellom Result1 og Result2,

b) Result1 er den riktige funksjonen, c) Result2 er den riktige funksjonen.

Forklar svaret ditt ved å referere til følgjande PDDL i samband med blokk-world problemet.

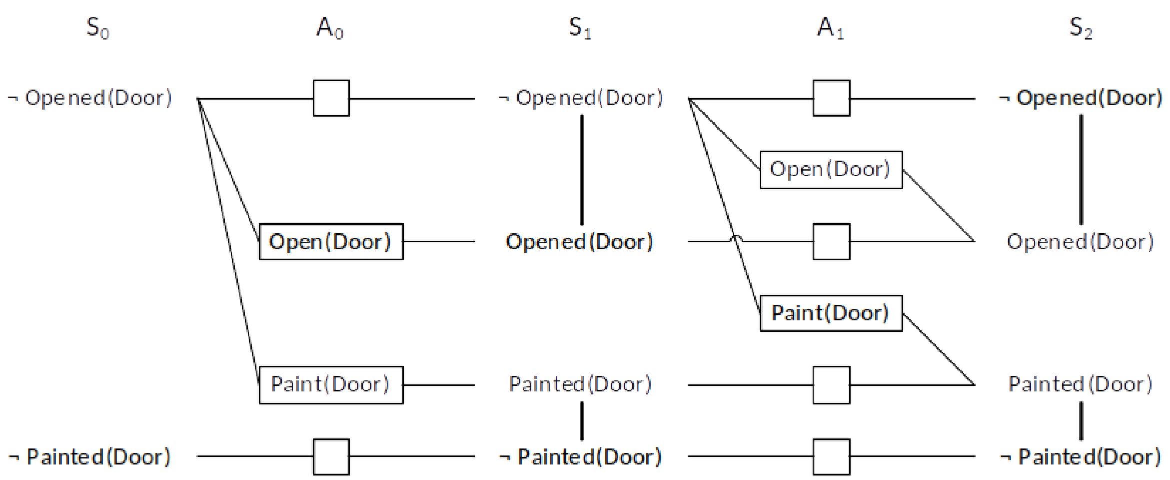
Action (MoveToTable(b, x)
PRECOND: On(b, x) ^ Clear(b) ^ Block(b) ^ (b ≠x)
EFFECT: On(b,Table) ^ Clear(x) ^ ¬ On(b, x))

Merk at i PDDL-"snippet" ovanfor så er **Add** og **Del** ikkje delt opp, men putta saman som **EFFECT**. Forklar kort svaret ditt.

B. For følgjande planlegginsproblem:

Init(¬Opened(Door) ^ ¬Painted(Door))
Goal(Opened(Door) ^ Painted(Door))
Action(Open(Door)
Precond: ¬Opened(Door)
Effect: Opened(Door))
Action(Paint(Door)
Precond: ¬Opened(Door)
Effect: Painted(Door))

har vi den følgjande "planning graph" kor nokre mutex-lenkjer manglar:



Finn dei manglande mutex lenkjar mellom tilstand sin-literals og handling si-literals på alle nivå. Skriv ned lista av manglande mutex-lenkjar for kvart nivå (både tilstand sin- og nivåa- til handling). For kvar mutex-lenkje, gje ei forklaring av hvofor dei er mutex. Svar ditt vil, til dømes, slik ut:

S1:

¬Opened(Door), Opened(Door) – one negates the other
¬Painted(Door), Painted(Door) – one negates the other

S2:

¬Opened(Door), Opened(Door) – one negates the other
¬Painted(Door), Painted(Door) – one negates the other

Du treng ikkje teikne grafen, berre liste dei para av tilstand sin- og handling si-literaler som har mutex-relasjonar mellom seg, saman med ei forklaring for kvart par som vist i dømet ovanfor.

SLUTT på OPPGÅVER
LUKKE TIL!

ENGLISH

Problem 1 [Agents. Answer 1(True) or 0 (False) for each of the following sentences.]

Note 2020: Several students have pointed out that there sometimes exists reasonable alternative answers to questions about agent environments. On future exams these kinds of questions will be discussion questions where you will explain your answers and what assumptions you make.

A. Simple reflex agents often use situation-actions rules for their reasoning. (T)
B. Simple reflex agents have an internal model of the current state of the world. (F)
C. An agent's utility function is essentially internalization of the performance measure.(T)
D. In order to be rational an agent should have a utility function. (F) Not necessarily. For ex, Simple vacuum cleaner does not have one but behaves rationally.
E. In a learning agent the learning element is responsible for selecting the external action to be executed next. (F)
F. Taxi driving happens in a stochastic, sequential, partially observable, and continuous environment. (T)
G. Agents need an internal model to cope with partially observable environments. (T)
H. An agent system has an agent function that selects the next action and a state transition function which computes what the next state of the environment would be when an action is executed.(T)
I. The action selection in model based agents uses as input the currently sensed data as well as the internal representation of the environment. (T)
J. In goal-based agents, rationality of the agents is measured based on the utility function. (F)

Problem 2 [General Questions about Search and Logic. Answer 1(True) or 0 (False) for each.]

a. Depth-first search is more space efficient than Breadth-first search. (True)
b. The basic Genetic Algorithm is Stochastic Beam Search supplemented with a crossover operator. (True)
c. Iterative Deepening search is more space efficient than Breadth-first search? (True)
d. One of the main reasons why Texas Hold'em poker is harder for AI than chess is that there are 52 cards in a deck, but only 32 pieces on a chessboard. (False)
e. Simulated Annealing is Best-First search supplemented with a temperature variable. (False)
f. All of the following are traditionally classified as Informed Search methods: A*, Iterative Deepening and Hill Climbing. (False)
g. To use a resolution theorem prover, all logical expressions must first be converted into horn-clause form. (False)
h. In first-order logic, a sentence is satisfiable if and only if there is at least one interpretation and one variable assignment in which it is true. (True)
i. PROLOG is a computer language that relies heavily on backward chaining. (True)
j. Skolemization is an important step of resolution theorem proving in both propositional logic and first-order logic. (False)

Problem 3

Assume the following predicates:

a. S(x) – x is a student
b. E(y) – y is an exam
c. Q(x) – x is a question.
d. EQ(x,y) – y is a question on exam x
e. K(x,y) – x knows the answer to question y.

and the following expressions:

1. $\forall x: S(x) \Rightarrow \{ \exists y, z: E(y) \wedge EQ(y,z) \wedge \neg K(x,z) \}$
2. $\forall x,y: E(x) \wedge EQ(x,y) \Rightarrow \{ \exists z: S(z) \wedge \neg K(z,y) \}$
3. $\forall x,y,z: E(x) \wedge Q(y) \wedge S(z) \Rightarrow \{ K(z,y) \Rightarrow EQ(x,y) \}$

Part 1:

For each logical expression, select the one natural-language sentence below that best captures its meaning:

A. There are some students who miss every question on all exams.
B. All exams have at least one question that no student can answer correctly.
C. Every student gets everything correct on at least one exam.
D. No student is perfect.
E. No exam question is correctly answered by every student.
F. Students do not know the answers to questions that are not on exams.
G. Given any student, question and exam, the student can answer the question if it is on the exam.
H. Each exam has a question that every student can answer correctly.

Answers:

1: D – Can be found by process of elimination or because a student not being perfect means with respect to the problem domain that they do not know the answer to a question.
2: E
3: F

Part 2:

One of the three logical sentences above yields the following expression when converted to Conjunctive Normal Form (CNF), where F and G are skolem functions:

$$\{\neg S(x) \vee E(F(x))\} \wedge \{\neg S(x) \vee EQ(F(x),G(x))\} \wedge \{\neg S(x) \vee \neg K(x,G(x))\}$$

Which one of the three logical sentences is it?

(Answer: #1)

Part 3:

What is the resolvent clause when the binary resolution rule is applied to the following two clauses (where F is a skolem function and Karen is a constant symbol)?

$\neg S(x) \lor E(F(x)) \lor Q(y)$
$S(Karen)$

Answer: E(F(Karen)) ∨ Q(y)

Problem 4

Figure 1 displays a search tree generated by Minimax search. Inside of each leaf node is its evaluation. Child nodes are always generated and evaluted from left to right in the tree.

List all of the leaf nodes that will **NOT** be generated if Minimax is run again, but this time with alphabeta pruning.

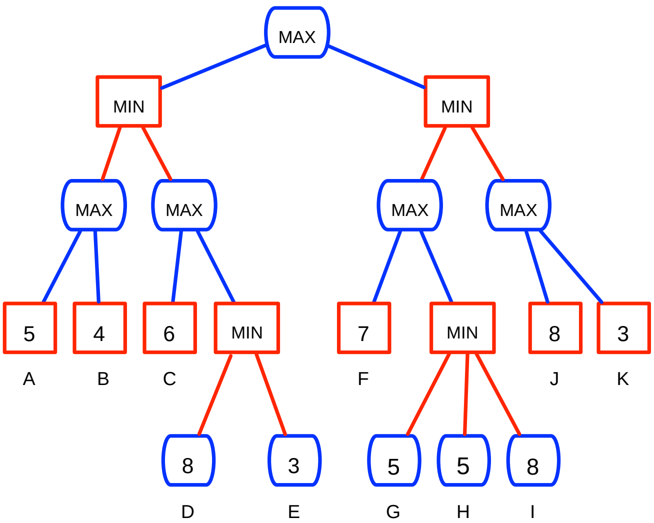


Figure 1: Minimax Search Tree

Answer: Nodes D, E, H, I, K

Problem 5

Figure 2 shows part of a tree built during A* search. The task is to rearrange the blocks to achieve the goal state while minimizing the total distance travelled by the blocks, where the width of each block has a distance = 1. The only legal operator is to switch the positions of two blocks. The heuristic is simply the total of the distances of all blocks from their goal locations.

Fill in all missing f, g and h values in the figure.

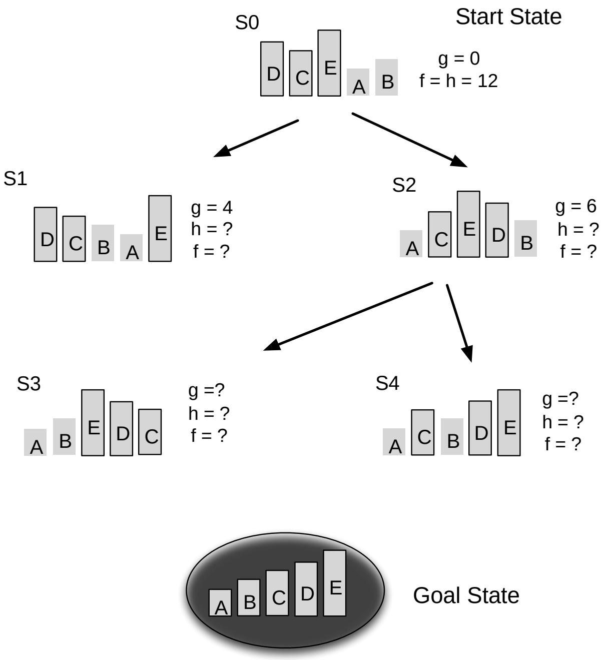


Figure 2: A* Search Tree

Answer: S1: h = 8, f = 12 S2: h = 6, f = 12
S3: g = 12, h = 4, f = 16 S4: g = 10, h = 2, f = 12

Problem 6

Figure 3 displays the use of model checking to test whether the following is a **valid** logical expression (using the formal definition of logical validity):

$$\{A \Rightarrow (B \Rightarrow C)\} \Rightarrow \{A \Rightarrow C\}$$

Fill in all missing cells of the table with a 1 (True) or 0 (False).
Next, based on the completed table, tell whether or not the expression is valid.

$$\{A \Rightarrow (B \Rightarrow C)\} \Rightarrow \{A \Rightarrow C\}$$

A	B	C	$B \Rightarrow C$	$A \Rightarrow (B \Rightarrow C)$	$A \Rightarrow C$
0	0	0	1	1	1
0	0	1	1	1	1
0	1	0	0	1	1

0	1	1	1	1	1
1	0	0	1	1 **	0 **
1	0	1	1	1	1
1	1	0	0	0	0
1	1	1	1	1	1

The two starred boxes prove that this is NOT valid.

Figure 3: Model Checking

Problem 7 [Constraint Satisfaction Problems (CSPs)]

Part 1:

Figure 4 shows a simple CSP involving 3 integer-valued variables and 3 constraints. The AC-3 (Arc consistency 3) algorithm is applied to the problem, and this involves several calls to the REVISE algorithm. The first 4 of these calls are shown. Fill in the updated domain for the variable listed after each call.

The right of the figure shows one complete solution to the CSP. Are there others? If so, list them.

Part 2:

Figure 5 shows a second CSP, with 3 integer-valued variables and 3 constraints. The solution method is local search, beginning with the state (3,3,3), and using the MIN-CONFLICTS algorithm to determine the best new value for each randomly-chosen variable. Assuming that the first variable chosen is Z, followed by Y, determine the missing values in the two states (2 and 3) shown in the figure. In this problem, a conflict is a constraint that is violated.

Variables & Domains

$X \in \{1, 2, 3, 4, 5\}$
$Y \in \{1, 2, 3, 4, 5\}$
$Z \in \{1, 2, 3, 4, 5\}$

Constraints

$2Y < X$
$X > 3Z$
$Y < 3Z$

1 REVISE(X,Y) X

$\in \{3, 4, 5\}$

2 REVISE(Z,X)

$Z \in \{1\}$

3 REVISE(Y,Z)

$Y \in \{1, 2\}$

4 REVISE(X,Z)

$X \in \{4, 5\}$

Solutions

X	Y	Z
4	1	1
5	1	1
5	2	1

Figure 4: AC-3 and REVISE

Variables & Domains

$X \in \{1, 2, 3, 4, 5\}$
$Y \in \{1, 2, 3, 4, 5\}$
$Z \in \{1, 2, 3, 4, 5\}$

Constraints

$X > Y + 1$
$X + 1 = Z$
$Z \geq 2Y$

1

X	**Y**	**Z**
3	3	3

Reassign Z

2

X	**Y**	**Z**
3	3	**4**

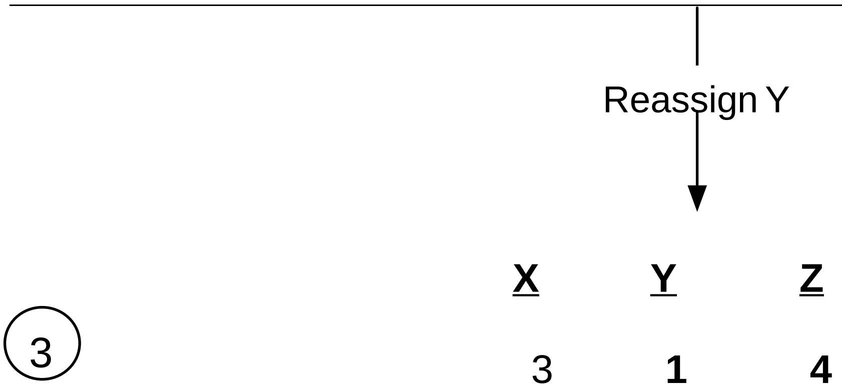


Figure 5: MIN-CONFLICTS

Problem 8 [Planning graphs]

Rob and Mary are at home. Rob has to get to work and has two options: either he walks or he uses the car. Mary needs to get to the airport and her only option is to use the car. The car needs fuel before it can be used. For the sake of brevity we will used the following simple string description for the states and actions.

robAtHome	Rob is at home
maryAtHome	Mary is at home
fuel	there is fuel in the car
car	car is at home

The actions to be modelled are:

```
Walk {
        precond: robAtHome
        add: robAtWork
        del: robAtHome
}

GetFuel {
        precond:
        add: fuel
        del:
}
DriveAirport{
         precond: maryAtHome, fuel, car
|         add: maryAtAirport
          del:maryAtHome, car
    }

DriveWork{
        precond: robAtHome, fuel, car
        add: robAtWork
        del:robAtHome, car
    }
```

The goal : robAtWork, maryAtAirport
The initial condition: robAtHome, maryAthome, car

Figure 6 illustrates the incomplete planning graph. The dashed lines at A0 level show the mutex relations. The gray boxes mean ”no operation”.

Part 1:

Mark the following mutex relationships either ”correct” or ”wrong” according to whether they are mutex at stage A1 or not, respectively. It is recommended that you find the mutex relations at S1 first. In the following list “robAtHome” and the other literal symbols refer to the persistence actions of the literals.

A. (Walk, robAtHome)
B. (Walk, robAtWork)
C. (DriveAirport, DriveWork)
D. (DriveAirport, car)
E. (robAtWork, DriveWork)
F. (robAtWork, ¬robAtWork)
G. (maryAtAirport,car)
H. (¬fuel, DriveWork)
I. (DriveWork, ¬robAtHome)
J. (DriveWork,fuel)

Part 2:

Is there a need to expand the graph after S2? Explain why (not) with 1 (or max 2) sentences.

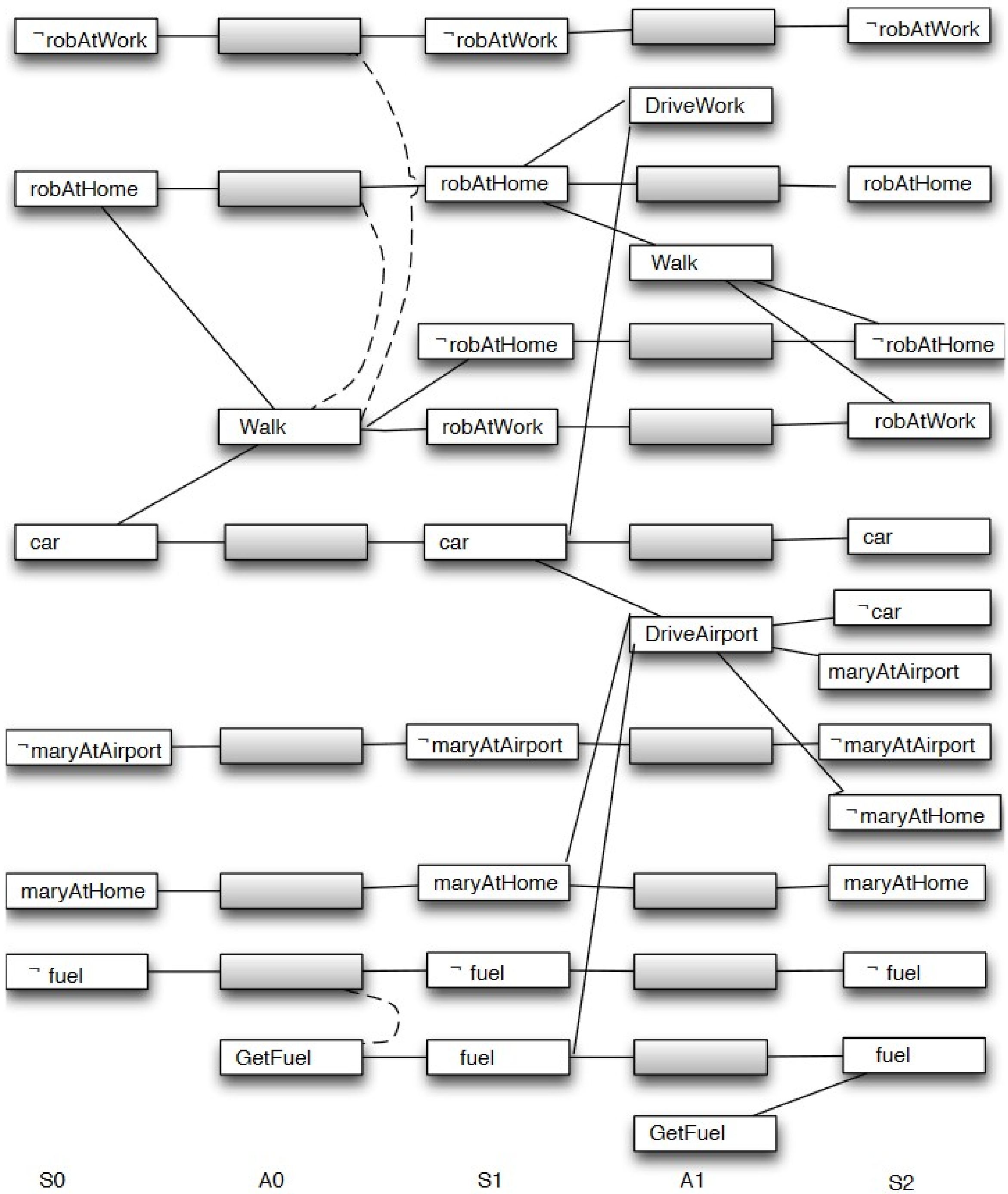


Figure 6 Planning graph for Problem 8

ANSWER:

Her er utskrift av alle lag med mutex relasjoner:

S0

-robAtWork
car
robAtHome
-maryAtAirport
maryAtHome
-fuel

A0

robAtHome
-maryAtAirport
car Walk
maryAtHome
-fuel
GetFuel
-robAtWork

mutex: (Walk,-robAtWork)
(Walk,robAtHome)
(-fuel,GetFuel)

S1

-robAtHome
-robAtWork
robAtWork
car
fuel
robAtHome
-maryAtAirport
maryAtHome
-fuel

mutex:
(-robAtHome,-robAtWork)
(-robAtHome,robAtHome)
(-robAtWork,robAtWork)
(robAtWork,robAtHome)
(fuel,-fuel)

A1:
robAtHome

DriveAirport
-maryAtAirport
car
-robAtHome
fuel
Walk
maryAtHome
DriveWork
robAtWork
-fuel
GetFuel
-robAtWork

mutex:
(Walk,-robAtWork)
(Walk,-robAtHome)
(Walk,DriveWork)
(Walk,robAtHome)
(Walk,robAtWork)
(DriveAirport,DriveWork)
(DriveAirport,-fuel)
(DriveAirport,-maryAtAirport)
(DriveAirport,maryAtHome)
(DriveAirport,car)
(robAtWork,DriveWork)
(robAtWork,-robAtWork) (robAtWork,robAtHome)
(-fuel,DriveWork)
(-fuel,fuel)
(-fuel,GetFuel)
(DriveWork,-robAtWork)
(DriveWork,-robAtHome)
(DriveWork,robAtHome)
(DriveWork,car)
(-robAtWork,-robAtHome) (robAtHome,-robAtHome)

S2

-car
-robAtHome
-robAtWork
robAtWork
maryAtAirport
car
fuel
-maryAtHome

robAtHome
-maryAtAirport
maryAtHome
-fuel

mutex:
(-car,car)
(-car,-fuel)
(-robAtHome,-robAtWork)
(-robAtHome,robAtHome)
(-robAtWork,robAtWork)
(robAtWork,robAtHome)
(maryAtAirport,car)
(maryAtAirport,-maryAtAirport)
(maryAtAirport,maryAtHome)
(maryAtAirport,-fuel)
(car,-maryAtHome)
(fuel,-fuel)
(-maryAtHome,-maryAtAirport)
(-maryAtHome,maryAtHome)
(-maryAtHome,-fuel)

Alle mål I S2 er oppnådd. Ingen expansion etter det.

Problem 9 [Knowledge representation languages.]

Part 1:

How would a Question Answering system answer the question ”What is the color of bird?” if it operates on the semantic network shown in Figure 7?

A. It is white.
B. It is yellow.
C. It is red.
D. It is red or white or blue.
E. Don’t know .

ANSWER: E ”I don’t now” – ”bird” har ikke farge attribute , heller ikke animal.

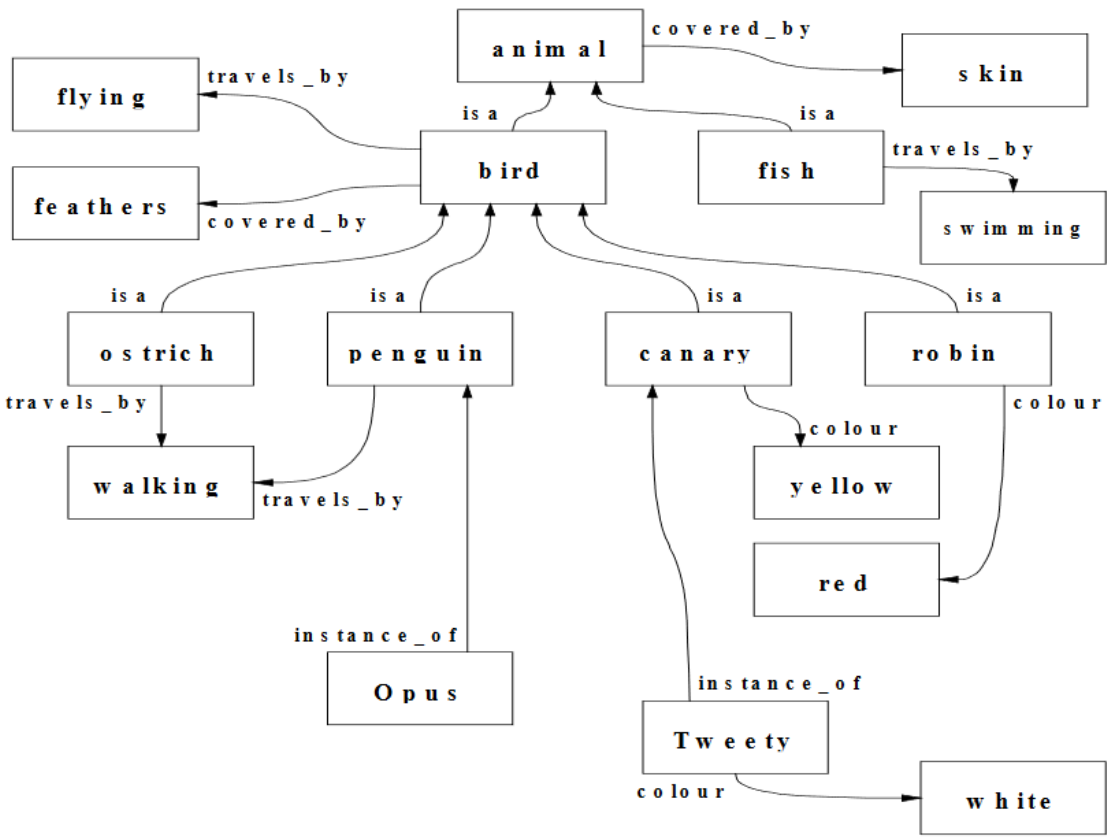


Figure 7. Semantic Network for "bird"

Part 2:
Which of the following is (are) true for the knowledge base shown in **Figure 8**? A. Apple-1 weighs 10 gram.
B. Apple-1 weighs 50 gram.
C. Apple-1 weighs 100 gram.
D. Apple-1 weighs 200 gram.
E. Apple-1 is green or red.

ANSWER: E, Apple-1 is green or red

Part 3:
You will represent the sentence *"Jack kidnapped Billy on August 5"* be as an event in a frame-based language. Write down the frame.

ANSWER:

Kidnap1

Is-a: Kidnapping event
perpetrator: Jack
victim: Billy
date: August 5

Ikke nødvendigvis disse attribute navnene trenges. Noe som har same funkjon/rolle blir/telles riktig.

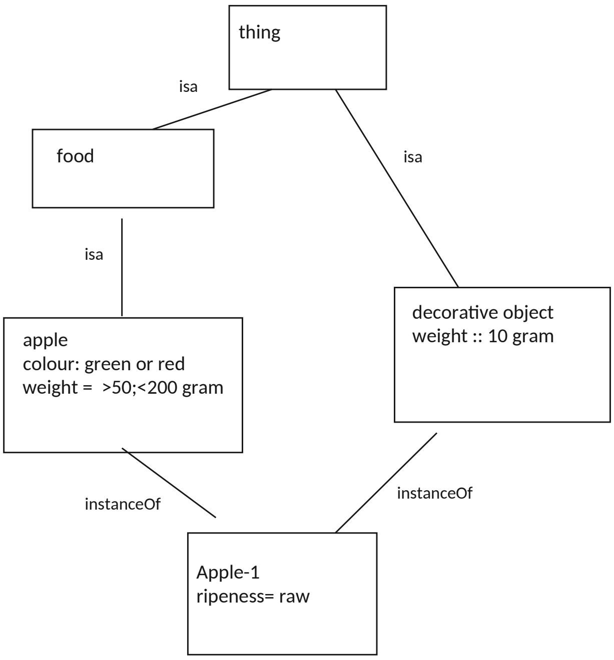


Figure 8 Apple model

Problem 10 [General, mixture of chapters 10-13 &22 and the ”An Overview of Knowledge Acquisition” (Musen) paper].

Fill in the blanks in the following questions:

A. You are a knowledge engineer and are working on a problem where the system’s reasoning mechanism relies on prototypical objects in the world.
”…FRAME/FRAME-BASED lang.….” would be the most appropriate representation language to use for this problem.

B. Sentiment analysis is a form of text ”…CATEGORIZATION………………….”.

C. Knowledge elicitation (as described in Musen’s paper) involves a knowledge engineer and a “ DOMAIN…………” expert.

D. Taste, smell and color of a cake are its ”…INTRINSIC…..……….” properties while weight and shape are its ”…EXTRINSIC………….” properties.

E. ”…PARTITIONS……………..” are both disjoint and exhaustive decomposition of categories.

F. In natural language processing, information ”…EXTRACTION…………..” is the process of acquiring knowledge from text.

G. ”…TURING………” test is the most known scenario for testing the intelligence of an articial intelligence system.

H. ”…PRECISION……………” is a measure used to evaluate IR systems and measures the proportion of documents in the result set that are actually relevant.

I. In frame-based languages, default reasoning is facilitated by inheritance of ”…ATTRIBUTE/SLOT………..” values.

J. In rule based systems, a depth first approach can be implemented by using ”RECENCY………..” as the conflict resolution strategy when selecting between the candidate rules that can fire at a certain time point.

END of QUESTIONS

GOOD LUCK!