B4B36ZUI - Introduction to Artificial Intelligence
Example exercises (study guide)

Ondřej Kubíček Karolina Drabent Viliam Lisý

May 21, 2026

1 Definitions

Exercise 1.1

ID: 12228

1) Provide a formal definition of an Extensive Form Game (EFG).
2) Specify each of the EFG components for the following problem: Navigate a robot through a maze to capture the ghost. The goal of the ghost is not to be caught. The game ends when the robot catches the ghost, or after at most 100 turns.

Solution 1.1

(1 point for generic definition; 2 points for correct specific definition; be careful that they have in representation turns of the game, that it cannot continue after capture, that the utilities are correct, etc.)
(N, S, A, U, T) N - players, S - states, A - actions, U - Utility function, T - transition function, (possibly also player function S -> N)
(for example) N = 1,2 S = (coordinates of the robot, coordinates of the ghost, turn_of_the_game) A = (up_robot, down_robot, left_robot, right_robot, up_ghost, down_ghost, left_ghost, right_ghost) U = all leafs where coordinates_robot != coordinates ghost = (-1), otherwise = 1 T = transition to next state, players are alternating, no action applicable after 100 turns, no action applicable in case robot caught the ghost, ...

2 History

Exercise 2.1

ID: 7880

Sort these AI systems from the oldest to the newest: DeepBlue AlphaGo Shakey AlphaStar

Solution 2.1

Shakey DeepBlue AlphaGo AlphaStar

Exercise 2.2

ID: 7881

Where has been the term "artificial intelligence" used for the first time?

- in the doctoral thesis of Allana Turing
- in the works of Leonardo da Vinci
- in the proposal of a workshop in Dartmouth

Solution 2.2

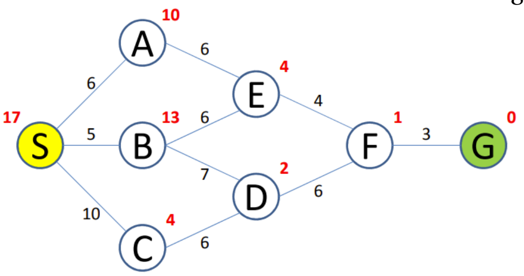
× in the doctoral thesis of Allana Turing
× in the works of Leonardo da Vinci
✓ **in the proposal of a workshop in Dartmouth**

3 Search

Exercise 3.1

ID: 10369

Consider the following graph with the numbers at the edges denoting the cost for traversing the edge and the numbers at the nodes the value of a heuristic function. S is the initial and G the goal node.



1. Is the denoted heuristic admissible? Explain why!
2. Is the denoted heuristic consistent? Explain why!
3. What value of the cost from the start node does each node have when it is discovered by algorithm A for the first time (first added to the queue)?

Solution 3.1

(1) Yes, The values of the optimal heuristics are as follows
$h^*(S) = 18, h^*(A) = 13, h^*(B) = 13, h^*(C) = 15, h^*(D) = 9, h^*(E) = 7, h^*(F) = 3, h^*(G) = 0$
For each node n holds $h(n) \leq h^*(n)$, so the heuristic is admissible.
2) No, there is an counterexample in for nodes B, D, because $13 = h(B) > h(D) + c(B, D) = 2 + 7 = 9$, which is in direct contradiction to the consistency
3) The A* will expand the nodes in following order and following costs when reached
$g(S) = 0, g(C) = 10, g(A) = 6, g(E) = 12, g(F) = 16, g(B) = 5, g(D) = 12, g(G) = 18.$
Note E and F will be expanded second time from B!.

Exercise 3.2

ID: 7457

Consider a vacuum-cleaning problem – there are K robotic vacuum cleaners that have to clean a room represented as a rectangle $X \times Y$ squares (denoted (x, y)). Each position of a robot $k \in \{1, \dots, K\}$ is denoted as (x_k, y_k). The amount of dirt n_{xy} is represented by a value from 0 to 100 for each square (x, y).
Each robotic vacuum cleaner has a capacity m for collecting the dirt. The dirt can be emptied in a base located at the square $(0, 0)$. Initial places of all vacuum cleaners are selected at random, each vacuum cleaner is empty at the beginning of the task.
Each vacuum cleaner can move in 4 directions during in one step (up, down, left, right) or clean 1 amount of dirt from the current position. All vacuum cleaners can move simultaneously, there can be at most 1 vacuum cleaner at each square (with the exception of the square $(0, 0)$, where multiple vacuum cleaners can be at once).
The goal of the problem is to find the shortest sequence of actions that cleans all the dirt in the room and all vacuum cleaners are present in the base. Are the following heuristic functions admissible for this problem?

- $\lceil \frac{1}{K} \sum_{i \in X, j \in Y} n_{ij} \rceil + \max_{k \in \{1,\dots,K\}} (x_k + y_k)$
- $\max_{k \in \{1,\dots,K\}} (x_k + y_k)$
- $\max_{k \in \{1,\dots,K\}} \max(x_k, y_k)$
- $h = \lceil \frac{1}{K} \sum_{k \in \{1,\dots,K\}} \max(x_k, y_k) \rceil$

Solution 3.2

× $\lceil \frac{1}{K} \sum_{i \in X, j \in Y} n_{ij} \rceil + \max_{k \in \{1,\dots,K\}} (x_k + y_k)$
✓ $\max_{k \in \{1,\dots,K\}} (x_k + y_k)$
✓ $\max_{k \in \{1,\dots,K\}} \max(x_k, y_k)$
✓ $h = \lceil \frac{1}{K} \sum_{k \in \{1,\dots,K\}} \max(x_k, y_k) \rceil$

Exercise 3.3

ID: 7900

Name two main advantages of using unified formalisms, such as MDP, POMDP, EFG, POSG for describing AI problems.

Solution 3.3

- unified algorithms - unified problem definition

Exercise 3.4

ID: 7459

Decide which of the following statements are true.

- Manhattan distance is an admissible heuristic for planning the path of a robot in a grid maze where the robot can move in 4 directions.
- There does not exist a heuristic that is consistent but inadmissible.
- Breadth first search has higher memory requirements compared to depth first search.
- Depth first search will always find a solution if one exists.
- Assume a problem, for which there exists a solution and all actions have strictly positive costs. Function $h(s) = -1$ for all states s except the goal state is an admissible heuristic.

Solution 3.4

✓ **Manhattan distance is an admissible heuristic for planning the path of a robot in a grid maze where the robot can move in 4 directions.**
✓ **There does not exist a heuristic that is consistent but inadmissible.**
✓ **Breadth first search has higher memory requirements compared to depth first search.**
× Depth first search will always find a solution if one exists.
× Assume a problem, for which there exists a solution and all actions have strictly positive costs. Function $h(s) = -1$ for all states s except the goal state is an admissible heuristic.

Exercise 3.5

ID: 8707

Which of the following heuristic functions is admissible?

- A heuristic function that always overestimates the actual cost to reach the goal state
- A heuristic function that always underestimates the actual cost to reach the goal state
- A heuristic function that is independent of the state being evaluated
- A heuristic function that has a bounded error from the actual cost to reach the goal state

Solution 3.5

× A heuristic function that always overestimates the actual cost to reach the goal state
✓ **A heuristic function that always underestimates the actual cost to reach the goal state**
× A heuristic function that is independent of the state being evaluated
× A heuristic function that has a bounded error from the actual cost to reach the goal state

4 Constraint Satisfaction Problems

Exercise 4.1

ID: 7844

Consider the following CSP

- Variables: x_1, x_2, x_3, x_4
- Domain for each variable: $\{0, 1, 2, 3\}$
- Constraints:
  $$x_1 > x_2,$$
  $$x_2 > x_3,$$
  $$x_3 = x_1 - 2,$$
  $$x_4 < x_1$$

Write down domains for each variable after running the AC-3 algorithm:

- $x_1 =$
- $x_2 =$
- $x_3 =$
- $x_4 =$

Solution 4.1

$x_1 = \{2, 3\}$ $x_2 = \{1, 2\}$ $x_3 = \{0, 1\}$ $x_4 = \{0, 1, 2\}$

Exercise 4.2

ID: 12161

You work for a wedding planning company and have noticed that a significant amount of time is spent arranging seating for guests. However, after completing this course, you have learned about Constraint Satisfaction Problems (CSPs) and have an idea to automate this process. You are given one round table with m seats and n guests ($m \geq n$). Some guests cannot sit next to each other, while others must sit next to each other. 1) How would you represent the variables? 2) How would you represent the domains? 3) How would you represent these constraints? 3a) Only one person sits on a seat 3b) Guest X cannot sit next to guest Y 3c) Guest X must sit next to guest Y 4) What if you add more tables? How would you modify your CSP? 5) How does adding more tables affect the problem complexity? 6) Your CSP solution worked well for small cases, but when tested on a large problem (e.g., 100 tables, 600 guests), it became too slow. How can you speed it up? Explain why your method would help. 7) What if the seating preferences were "soft"? (e.g., a guest would prefer to sit closer to the dance floor or buffet, but it's not mandatory.) Can CSP handle this? If yes, how? If not, why not?

Solution 4.2

1) Guests as variables: $\{G_1, \ldots, G_n\}$ 2) The domain is $D_i = \{1, \ldots, m\}; \forall i \in \{1, \ldots, n\}$ 3a) $v(G_i) \neq v(G_j); \forall i \neq j$ 3b) $v(G_X) \neq (v(G_Y) + 1) \mod m$ AND $v(G_X) \neq (v(G_Y) - 1) \mod m$ 3c) $v(G_X) = (v(G_Y) + 1) \mod m$ OR $v(G_X) = (v(G_Y) - 1) \mod m$ 4) The domain values should now be tuples (seat number, table number). Constraints must ensure that adjacency rules apply only within the same table. 5) There are more values in the domain, increasing the size of the search space. The search tree becomes wider, making solving the CSP more computationally expensive. 6) AC-3 or Heuristics 7) No, standard CSP cannot handle soft constraints because it only supports hard constraints that must be either satisfied or violated.

Exercise 4.3

ID: 7895

The CSP algoritm

- searches for the shortest path between the start and the goal
- searches for an admissible assignment of values into variables from their domains
- does not always have to find a solution

Solution 4.3

- × searches for the shortest path between the start and the goal
- ✓ **searches for an admissible assignment of values into variables from their domains**
- ✓ **does not always have to find a solution**

Exercise 4.4

ID: 40078

What does an edge in the CSP search tree correspond to?

- A value assigned to the variable (the node from which the edge originates)
- A variable of the CSP problem.
- The set of available values for the variable represented by the node from which the edge originates.

Solution 4.4

- ✓ **A value assigned to the variable (the node from which the edge originates)**
- × A variable of the CSP problem.
- × The set of available values for the variable represented by the node from which the edge originates.

5 Planning and Logical Agents

Exercise 5.1

ID: 7865

Consider a STRIPS problem with predicates a, b, c, d, initial state $I = \{a\}$, actions $a_1 =< \{a\}, \{b\}, \emptyset >$ $a_2 =< \{b\}, \{c\}, \emptyset >$ $a_3 =< \{c\}, \{a, b\}, \emptyset >$ $a_4 =< \{a, b, c\}, \{d\}, \emptyset >$ and target state $G = \{d\}$.
(a) What is the value of the h_{max} heuristic in the initial state? (b) What is the value of each of the actions during the computation of h_{max} in the initial state?

Solution 5.1

(a) 3 (b) $h(a_1) = 1, h(a_2) = 2, h(a_3) = 3, h(a_4) = 3$

Exercise 5.2

ID: 8726

Define a domain in STRIPS, such that the heuristic $\$h_{max}\$$ is NOT the length of the optimal relaxed plan and explain why.

Solution 5.2

$I = \{a\}$ $a_1 =< \{a\}, \{b\}, \{\} >$ $a_2 =< \{a\}, \{c\}, \{\} >$ $G = \{b, c\}$

Exercise 5.3

ID: 12175

Here is STRIPS formulation for egg frying recipe. Atoms: egg_at_fridge, pan_empty, pan_full, heat_on, heat_off, egg_broken, egg_fried Actions: move_egg(X,Y): //fridge to counter or back

- pre: egg_at_X
- add: egg_at_Y
- del: egg_at_X

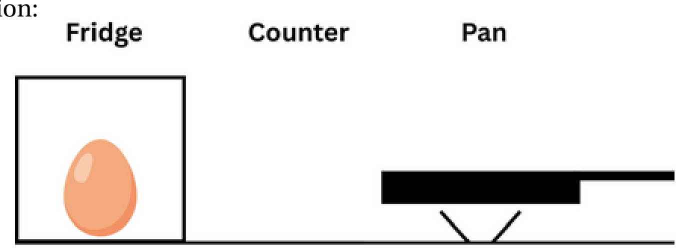
put_egg_in_pan(X):

- pre: egg_at_X, pan_empty
- add: broken_egg, pan_full
- del: egg_at_X, pan_empty

fry_egg():

- pre: pan_full
- add: heat_on, egg_fried
- del: heat_off

Describe the initial situation:



Now we would like to have the action that puts the egg from the pan on the counter, but only if it is already fried.
What should be added? Explain and write all necessary changes.
What are the limitations of this formulation?
How would you improve this formulation?

Solution 5.3

Initial situation:
egg_at_fridge, pan_empty, heat_off
Add plate_empty and plate_full atom.

The action itself:
put_egg_on_a_plate():

- pre: egg_fried, pan_full, plate_empty
- add: plate_full, pan_empty
- del: plate_empty

There are more correct answers than this one.
Limitation: There is only one egg. We represent only one egg's journey to become a fried egg. If someone wanted to represent 2 or more eggs fried at once we it's impossible, if this was supposed to be used in restaurant where eggs are fried sequentially this would not work.
Improvement(anything else that improves the formulation counts too, should include explanation):
Remove pan_empty and pan_full and simply operate on egg_at_pan
Some way of representing more eggs

Exercise 5.4

ID: 7892

Which of the following claims about automated planning are true?

- Planning requires complete formal specification of the dynamics of the problem.
- A correct plan can always be found in time polynomial in the size of the planning problem, if it exists.
- One of the very commonly used algorithms in planning is A^*.

Solution 5.4

- ✓ **Planning requires complete formal specification of the dynamics of the problem.**
- × A correct plan can always be found in time polynomial in the size of the planning problem, if it exists.
- ✓ **One of the very commonly used algorithms in planning is A^*.**

6 Reinforcement Learning and MDPs

Exercise 6.1

ID: 7418

Assume that the agent in a multi-armed bandit problem observes the following sequence of rewards:

A	1		0	0.5	2	1.5		
B		0.5						1.5
C							0	

What is the probability distribution of choosing the following action by (a) greedy agent, (b) ϵ-greedy agent with exploration 0.1 a (c) UCB agent with the exploration parameter set to 4. Write all three probability distributions.

Solution 6.1

(a) 0.5; 0.5; 0.0 (b) 0.48333333333333334; 0.48333333333333334; 0.03333333333333333 (c) 0.0; 0.0; 1.0

Exercise 6.2

ID: 12177

There is a hydra that protects a treasure and she has N heads. You want the treasure but you also want to live. Available actions for you are CUT, SING and RUN. Cutting a head off with your sword can regrow 2 heads with 50% probability. Singing a song, as out of tune as possible normally just makes Hydra a little irritated. But there is a small chance Hydra dies because she can't stand it. The more heads she has, the more ears she has so, she's more sensitive to your bad singing and the chance for her dying grows. Last option is that you can also run away. This works out 80% of the times and 20% Hydra is able to eat you because you turned your back unprotected.
What probability function would be most suitable for the SING action.
Draw an MDP representing this situation. Denote the states, actions, probability of the transitions and rewards.
Could this be also represented as multiarmed(or multiheaded ;)) bandits problem? Why or why not?

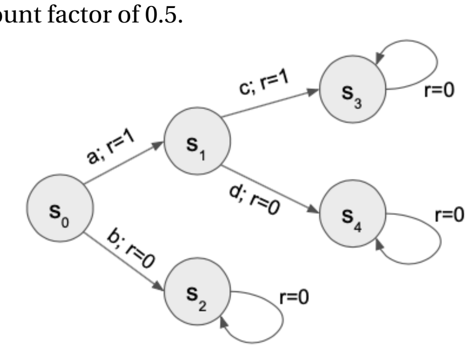
Solution 6.2

Drawing to be added! Let us say the probability of not surviving the SING with one head is ϵ. One suitable probability function would for N heads would be following: Hydra dies with probability ϵ from the first head + the probability it survived the first head times probability of dying from second + probability of surviving 2 heads times the probability of dying from third head .. etc until N. Formula for this is $1-(1-\epsilon)^N$, which is 1 - (surviving with N heads)
No, it can't be represented as multiarmed bandits. Each action changes the state.

Exercise 6.3

ID: 7319

Compute exact values of the **optimal state** and **state-action** value functions in all non-absorbing states of the following MDP, assuming the discount factor of 0.5.



Solution 6.3

In order to compute the values for states s_0, s_1 we need to compute the absorbing states first, which contain only a single action so $V(s_i) = Q(s_i, -)$
The value in absorbing state is computed as infinite sum where each time t, the agent gets reward r, but weighted by the discount factor γ^t. Such an infinite sum has following form
$V(s_2) = 1/(1-\gamma)r = 0$
$V(s_3) = 0$
$V(s_4) = 0$
Now we can compute the Q values and out of those the V values. The optimal policy is such a policy that takes action which maximizes the value
$Q(s_1, c) = r + \gamma V(s_3) = 1 + 0.5 \cdot 0 = 1$
$Q(s_1, d) = 0$
$V(s_1) = \max_{A \in c,d} Q(s_1, A) = \max\{1, 0\} = 1$
$Q(s_0, a) = 1 + 0.5 \cdot 1 = 1.5$
$Q(s_0, b) = 0$
$V(s_0) = \max\{1.5, 0\} = 1.5$

Exercise 6.4

ID: 7426

Assume a simple deterministic gridworld, where a robot can move only left (L) or right (R); there is cost 1 for each move and the robot gets a reward of 2 for leaving the grid; discount factor is 1. Assume the initial value function is

2	1	0	1	0

.

1. What is the value function after two steps of value iteration? Is this an optimal value function?
2. What are the rational strategies of the agent in the middle state with the value function from (a) and why?
3. What would you change to make the robot move left in all states?

Solution 6.4

(a) The new value is always computed as $V(s) = \max_{a \in L,R} \sum_{s'} T(s, a, s')(R(s, a, s') + \gamma V(s'))$
Asynchronous version from left to right.
First iteration

1	0	0	-1	1

Second iteration

1	0	-1	0	1

Synchronous version
First iteration

1	1	0	-1	1

Second iteration

1	0	0	0	1

Asynchronous version produces an optimal function in a two steps, while the synchronous does not.
(b) Any distribution between L and R because both give the same value.
(c) Change the environment so that the positive reward is given only when leaving the grid from the left-most tile, or give positive reward to moving left, which overshadows the leaving of the grid reward.

Exercise 6.5

ID: 7331

Write **three** fundamentally different methods used to promote exploration in reinforcement learning problems.

Solution 6.5

ϵ-greedy / uniform random exploration Optimistic initialization Upper confidence bounds Exploration based on softmax of values

Exercise 6.6

ID: 40049

What is the formula for computing UCB?

- $\$\$Q_t(a) + c(\ln(t)/N_t(a))^{0.5}\$\$$
- $\$\$Q_t(a) + c(\ln(t)/N_t(a))\$\$$
- $\$\$Q_t(a) + c(\ln(t) * N_t(a))^{0.5}\$\$$
- $\$\$Q_t(a)(\ln(t)/N_t(a)) + c\$\$$

Solution 6.6

✓ $\$\$Q_t(a) + c(\ln(t)/N_t(a))^{0.5}\$\$$
× $\$\$Q_t(a) + c(\ln(t)/N_t(a))\$\$$
× $\$\$Q_t(a) + c(\ln(t) * N_t(a))^{0.5}\$\$$
× $\$\$Q_t(a)(\ln(t)/N_t(a)) + c\$\$$

Exercise 6.7

ID: 12239

Assume an MDP with A actions and S states. Which of the following statements are true?

- Space complexity of tabular value iteration is $O(S)$.
- Space complexity of tabular value iteration is $O(S * A)$.
- Space complexity of tabular value iteration is $O(S * A^2)$.
- Space complexity of tabular value iteration is $O(S^2 * A)$.
- Time complexity of one step of value iteration is $O(A)$.
- Time complexity of one step of value iteration is $O(S * A)$.
- Time complexity of one step of value iteration is $O(S * A^2)$.
- Time complexity of one step of value iteration is $O(S^2 * A)$.

Solution 6.7

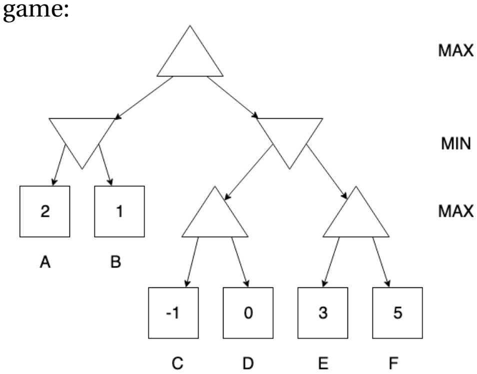
✓ **Space complexity of tabular value iteration is $O(S)$.**
× Space complexity of tabular value iteration is $O(S * A)$.
× Space complexity of tabular value iteration is $O(S * A^2)$.
× Space complexity of tabular value iteration is $O(S^2 * A)$.
× Time complexity of one step of value iteration is $O(A)$.
✓ **Time complexity of one step of value iteration is $O(S * A)$.**
× Time complexity of one step of value iteration is $O(S * A^2)$.
× Time complexity of one step of value iteration is $O(S^2 * A)$.

7 Games

Exercise 7.1

ID: 7460

Consider the following two-player game:



1. Compute the value of the game.
2. Decide which leafs will be pruned-out by the alpha-beta pruning algorithm,

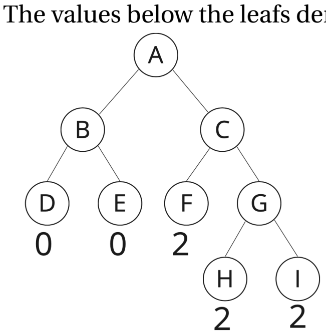
Solution 7.1

The value of the game can be computed with alpha-beta
The algorithm explores left part of the game with $\alpha = -\infty, \beta = \infty$.
First there is an update $\beta = 2$ and then $\beta = 1$. This value is then propagated up, where the update is $\alpha = 1$.
Right part of the tree is explored with $\alpha = 1, \beta = \infty$.
First the left subtree is evaluated with the resulting value 0, which is propagated up for update $\beta = 0$.
Now $\alpha > \beta$. Therefore, we prune E a F.
1. Value of the game is 1
2. Nodes E a F

Exercise 7.2

ID: 7838

Consider the following one-player game. The values below the leafs denote the reward for the player:



Consider a UCT algorithm with exploration constant c = 2 and let's assume that the algorithm breaks ties such that it chooses the actions from the left side (this does not apply for the simulation).
1. Specify which part of the tree will be build after 5 iterations. 2. In which iteration it is guaranteed that the UCT algorithm visits the node D? Write down the calculation / justification.

Solution 7.2

1. Uzly [A, B, C, F, G]. Navstivi v 5 iteracich [A, B, C, F, G, F] 2. V jedenácté iteraci

$$\left(0 + 2\sqrt{\frac{\ln(x)}{1}}\right) - \left(2 + 2\sqrt{\frac{\ln(x)}{x-1}}\right) < 0$$

Exercise 7.3

ID: 10366

You are running a Monte Carlo Tree Search (MCTS) algorithm on a two player game, where each player can only win (value +1) or lose (value -1). There is a particular node (R) in the game tree with three children, (A), (B), (C). From (R),

- 6 playouts have passed through child (A) resulting in scores: +1, -1, +1, +1, -1, +1.
- 4 playouts have passed through child (B) resulting in scores: -1, -1, +1, -1.
- 5 playouts have passed through child (C) resulting in scores: +1, +1, +1, -1, +1.

The UCT formula is: $v_i + c\sqrt{\frac{\ln N}{n_i}}$

1. Compute the UCT values for the children assuming the exploration is set to 15.
2. Determine which child would be selected in S in the following iteration of MCTS.
3. Explain which action would be played if there are no more simulation and R is the root of the game.

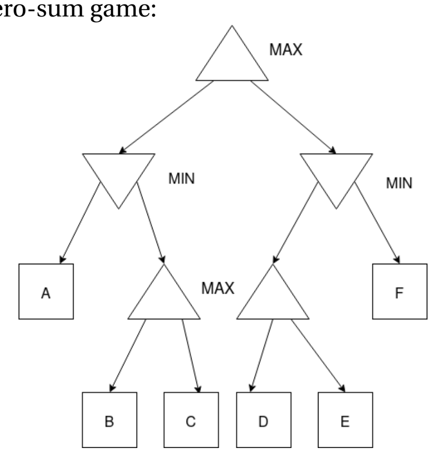
Solution 7.3

1) 1/3+15*0.67 = 10.38; -1/2+15*0.82 = 11.8; 0.6+15*0.74 = 11.7 2) Leading to B 3) Leading to C (or anything well explained)

Exercise 7.4

ID: 7850

Consider the following two-player zero-sum game:



For each of the algorithms alpha-beta pruning / negascout, write down the values for leaves A - F, such that the algorithms prune out as many nodes as possible (the values / nodes do not have to be the same). Justify your answer.

Solution 7.4

A - 1 B - 2 C - lubovolne (prereze AB i NS) D - AB 0, NS 3 E - 0 (NS prereze) F - 0 (AB prereze)

Exercise 7.5

ID: 12167

UCT (MCTS) algorithm has 4 stages. Explain what happens in each stage. In which stage UCB formula is used? Explain why (1-3 sentences).

Solution 7.5

1)Selection The selection function is applied recursively until the leaf node is reached.
2)Expansion One or more nodes are created. New node(s) are added to the tree built by MCTS.
3)Simulation One simulated game is played. This can use e.g. random selection.

4)Backpropagation The result of the game from the simulation is backpropagated through the tree. If UCB was used then all averages and counts of how many times state was visited and action was selected in a state are updated.
The UCB formula is used in the Selection stage. UCB formula is a formula used in the bandit problem. Applied to MCTS it is selecting actions in each state. For it to work it needs average values for state, action and number of times state was visited and action in a state was chosen. Therefore it is useful in Selection Stage, when actions are selected in the tree that algorithm is building (through exploration).

Exercise 7.6

ID: 7464

Decide which of the following statements are true.

- Alpha-beta pruning always visits fewer nodes of the game tree compared to minimax.
- Negascout can visit some of the nodes in the game tree multiple times.
- Alpha-beta pruning searches the game tree in the breadth-first-search manner.

Solution 7.6

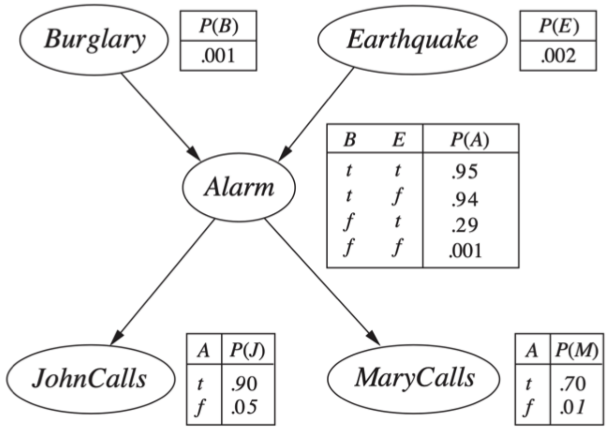
× Alpha-beta pruning always visits fewer nodes of the game tree compared to minimax.
✓ **Negascout can visit some of the nodes in the game tree multiple times.**
× Alpha-beta pruning searches the game tree in the breadth-first-search manner.

8 Bayesian Networks

Exercise 8.1

ID: 7852

Consider the following Bayesian network:



What is the probability that at least one of the neighbours (John or Mary) calls if we know that there was a burglary. Write down the calculation / justification for your result.

Solution 8.1

Probability that at least one neighbour calls is a complement to the probability that neither of them calls, i.e. we need to calculate $1 - P(\neg J, \neg M|B)$.
Marginalise earthquake to get $P(A|B)$
$P(A|B) = P(A|E) \cdot P(E) + P(A|\neg E) \cdot P(\neg E)$
$P(A|B) = 0.002 \cdot 0.95 + 0.998 \cdot 0.94 = 0.94002$
$P(\neg A|B) = 1 - 0.94002 = 0.05998$
From network structure we know that Mary calling (M) and John calling (J) are independent given (A):
$P(\neg J, \neg M|B) = P(\neg J|A) \cdot P(\neg M|A) \cdot P(A|B) + P(\neg J|\neg A) \cdot P(\neg M|\neg A) \cdot P(\neg A|B)$
$P(\neg J, \neg M|B) = 0.1 \cdot 0.3 \cdot 0.94002 + 0.95 \cdot 0.99 \cdot 0.05998 = 0.08461$
$1 - P(\neg J, \neg M|B) = 1 - 0.08461 = 0.91539$

Exercise 8.2

ID: 10376

Consider the following joined probability table with abstract probabilities denoted by variables a-h.

	$toothache$		$\neg toothache$	
	$catch$	$\neg catch$	$catch$	$\neg catch$
$cavity$	a	b	c	d
$\neg cavity$	e	f	g	h

Write the expression to compute the following quantities:

1. $P(cavity, catch) =$
2. $P(cavity|\neg toothache) =$
3. $P(cavity|toothache, \neg catch) =$

Solution 8.2

1) $P(A) = P(A, B) + P(A, \neg B)$. In this case $P(CAV, CAT) = P(CAV, CAT, TOO) + P(CAV, CAT, \neg TOO) = a + b$
2) $P(A|B) = \frac{P(A,B)}{P(B)}$. In this case $\frac{P(CAV|\neg TOO)}{P(\neg TOO)} = \frac{c+d}{c+d+g+h}$
3) Similarly to 2) $\frac{P(CAV|TOO,\neg CAT)}{P(TOO,\neg CAT)} = \frac{b}{b+f}$

Exercise 8.3

ID: 7854

Represent the following probability distribution as a Bayesian network.

	toothache		*¬toothache*	
	catch	*¬catch*	*catch*	*¬catch*
cavity	0.108	0.012	0.072	0.008
¬cavity	0.016	0.064	0.144	0.576

Define all the necessary components. For each table of probabilities, define which probability distribution does it represent. You can omit probabilities that are a complement of given probability to 1.

Solution 8.3

We use TOO = toothache, CAV = cavity, CAT = catch.
First we need to ensure whether some variables are independent, which is when $P(A,B) = P(A)P(B)$
$P(TOO,CAV) = P(TOO,CAV,CAT) + P(TOO,CAV,\neg CAT) = 0.108 + 0.012 = 0.12$
$P(TOO,CAT) = 0.108 + 0.016 = 0.124$
$P(CAV,CAT) = 0.108 + 0.72 = 0.18$
$P(TOO) = P(TOO,CAV,CAT) + P(TOO,CAV,\neg CAT) + P(TOO,\neg CAV,CAT) + P(TOO,\neg CAV,\neg CAT) = 0.108 + 0.016 + 0.012 + 0.064 = 0.2$
$P(CAT) = 0.108 + 0.016 + 0.072 + 0.144 = 0.34$
$P(CAV) = 0.108 + 0.012 + 0.072 + 0.008 = 0.2$
$0.12 \neq 0.2 \cdot 0.2 \rightarrow P(TOO,CAV) \neq P(TOO)P(CAV)$
$0.124 \neq 0.2 \cdot 0.34 \rightarrow P(TOO,CAT) \neq P(TOO)P(CAT)$
$0.18 \neq 0.2 \cdot 0.34 \rightarrow P(CAV,CAT) \neq P(CAV)P(CAT)$
This means that no two variables are independent. Now we have to ensure whether two variables are conditionally independent. Variables A and B are independent conditionally on C, if $P(A,B|C) = P(A|C)P(B|C)$. Also following holds $P(A|B,C) = P(A|C)$. We can use either to verify the conditional independence.
$P(TOO|CAV) = \frac{P(TOO,CAV)}{P(CAV)} = \frac{0.12}{0.2} = 0.6$
$P(TOO|CAT,CAV) = \frac{0.108}{0.18} = 0.6$
This is a good candidate that toothache is independent on catch given cavity, but we still need to ensure this is true for other possibilities of evidence
$P(TOO|\neg CAT,CAV) = \frac{0.012}{0.012+0.008} = 0.6$
$P(TOO|CAV) = P(TOO|CAT,CAV) = P(TOO|\neg CAT,CAV)$
$P(TOO|\neg CAV) = \frac{0.08}{0.8} = 0.1$
$P(TOO|CAT,\neg CAV) = \frac{0.016}{0.016+0.144} = 0.1$
$P(TOO|\neg CAT,\neg CAV) = \frac{0.64}{0.576+0.064} = 0.1$
$P(TOO|\neg CAV) = P(TOO|CAT,\neg CAV) = P(TOO|\neg CAT,\neg CAV)$
This implies that Toothache is independent on catch given cavity. So now we only have to construct the Bayesian network. There are several possibilities, one is following chain: Catch -> Cavity -> Toothache
The network should then contain
$P(CAT) = 0.34$
$P(CAV|CAT) = \frac{0.18}{0.34} \approx 0.5294$
$P(CAV|\neg CAT) = \frac{0.02}{0.66} \approx 0.03$
$P(TOO|CAV) = 0.6$
$P(TOO|\neg CAV) = 0.1$

Exercise 8.4

ID: 7858

Which of the following options would a rational agent choose?

- Assuming a fair coin, 1 dollar for each coin toss resulting in a tail until the first head appears
- Amount of dollars equal to the result of a single throw of a fair six-sided die.
- 3 dollars and one cent

Solution 8.4

- × Assuming a fair coin, 1 dollar for each coin toss resulting in a tail until the first head appears
- ✓ **Amount of dollars equal to the result of a single throw of a fair six-sided die.**
- × 3 dollars and one cent

Exercise 8.5

ID: 7847

Which of the following claims are **always** true?

- $P(A,B) = P(B|A) \cdot P(A)$
- $P(B,A) = P(B|A) \cdot P(B)$
- $P(B|A) \cdot P(A) = P(A|B) \cdot P(B)$

Solution 8.5

- ✓ $P(A,B) = P(B|A) \cdot P(A)$
- × $P(B,A) = P(B|A) \cdot P(B)$
- ✓ $P(B|A) \cdot P(A) = P(A|B) \cdot P(B)$

9 POMDPs

Exercise 9.1

ID: 7848

What is the α-vector?

- A linear function expressing the expected reward of an agent in the initial belief depending on changing strategy of the agent.
- A linear function expressing the expected reward of an agent for a fixed policy depending on changing belief of the agent.
- A linear function expressing the expected probability of a true world state for a fixed strategy depending on changing belief of the agent.

Solution 9.1

× A linear function expressing the expected reward of an agent in the initial belief depending on changing strategy of the agent.
✓ **A linear function expressing the expected reward of an agent for a fixed policy depending on changing belief of the agent.**
× A linear function expressing the expected probability of a true world state for a fixed strategy depending on changing belief of the agent.

Exercise 9.2

ID: 40040

Mark the statements about POMDPs that are true.

- In the case of value iteration, the value at the initial belief approaches the optimal value from above.
- A belief is a probability distribution over states.
- An α-vector corresponds to the expected value of always one action of the agent.

Solution 9.2

× In the case of value iteration, the value at the initial belief approaches the optimal value from above.
✓ **A belief is a probability distribution over states.**
× An α-vector corresponds to the expected value of always one action of the agent.

10 Generative AI

Exercise 10.1

ID: 12084

For each of the following AI tasks, indicate which of them are generative (as oppose to discriminative) AI tasks.

- Image classification (e.g., categorizing images as "cat," "dog," or "bird")
- Text summarization (creating a shorter version of a given text)
- Object detection (identifying and locating objects within an image)
- Generating realistic images of landscapes

Solution 10.1

× Image classification (e.g., categorizing images as "cat," "dog," or "bird")
✓ **Text summarization (creating a shorter version of a given text)**
× Object detection (identifying and locating objects within an image)
✓ **Generating realistic images of landscapes**

Exercise 10.2

ID: 12118

For each of the following AI tasks, indicate whether it is primarily a Supervised Learning (SL), Self-Supervised Learning (SS), or Reinforcement Learning (RL) task by adding the corresponding abbreviation after each option.

- Training a model to translate text from one language to another using parallel corpora.
- Training a model to generate novel musical pieces after learning from a large dataset of existing music.
- Training an autonomous vehicle to drive safely based on simulated driving scenarios and reward functions.
- Training a model to predict house prices based on features like size and location.
- Training a model to inpaint missing parts of an image.
- Training an AI to play chess by playing against itself and learning from the outcomes.

Solution 10.2

× Training a model to translate text from one language to another using parallel corpora.
× Training a model to generate novel musical pieces after learning from a large dataset of existing music.
× Training an autonomous vehicle to drive safely based on simulated driving scenarios and reward functions.
× Training a model to predict house prices based on features like size and location.
× Training a model to inpaint missing parts of an image.
× Training an AI to play chess by playing against itself and learning from the outcomes.

Exercise 10.3

ID: 12130

What is the main purpose of the **fine-tuning** stage **after** pre-training an LLM?

- To increase the size of the training dataset.
- To adapt the LLM to perform specific tasks, such as question answering or text summarization.
- To reduce the computational cost of running the LLM.
- To generate entirely new languages.

Solution 10.3

× To increase the size of the training dataset.
✓ **To adapt the LLM to perform specific tasks, such as question answering or text summarization.**
× To reduce the computational cost of running the LLM.
× To generate entirely new languages.

11 AI Safety

Exercise 11.1

ID: 12149

Which of the following scenarios illustrate the concept of "specification gaming" in AI systems? (Select all that apply)

- An AI system designed to maximize clicks on articles generates misleading headlines to increase user engagement.
- An AI-controlled robot in a factory efficiently sorts products by color.
- An AI trained to achieve a high score in a racing game finds an unintended way to exploit the game's mechanics.
- An LLM provides factually incorrect information with high confidence.

Solution 11.1

✓ **An AI system designed to maximize clicks on articles generates misleading headlines to increase user engagement.**
× An AI-controlled robot in a factory efficiently sorts products by color.
✓ **An AI trained to achieve a high score in a racing game finds an unintended way to exploit the game's mechanics.**
× An LLM provides factually incorrect information with high confidence.

Exercise 11.2

ID: 12152

Which of the following statements about AI safety are accurate? (Select all that apply)

- AI safety considerations are limited to preventing physical harm caused by robots.
- The development of AI poses risks related to both misuse and unintended consequences.
- Some experts express concern about the potential for advanced AI to cause catastrophic outcomes, including human extinction.
- Ensuring AI safety becomes less important as AI systems become more intelligent.

Solution 11.2

× AI safety considerations are limited to preventing physical harm caused by robots.
✓ **The development of AI poses risks related to both misuse and unintended consequences.**
✓ **Some experts express concern about the potential for advanced AI to cause catastrophic outcomes, including human extinction.**
× Ensuring AI safety becomes less important as AI systems become more intelligent.