The “solutions” to the exam/assignment problems in this document are supplied to give students a starting point for answering questions. While we are striving for helpful “solutions”, they can be incomplete and can even contain errors.

If you find “solutions” you do not understand or you find incorrect, discuss this on the course forum and/or with your TA and/notify the instructors.

In any case, grading student’s answers is not a process of simply “comparing with the reference solution”, therefore errors in the “solutions” are not a problem in this case.

In the course Artificial Intelligence I/II we award 5 bonus points for the first student who reports a factual error (please report spelling/formatting errors as well) in an assignment or old exam and 10 bonus points for an alternative solution (formatted in LaTeX) that is usefully different from the existing ones.

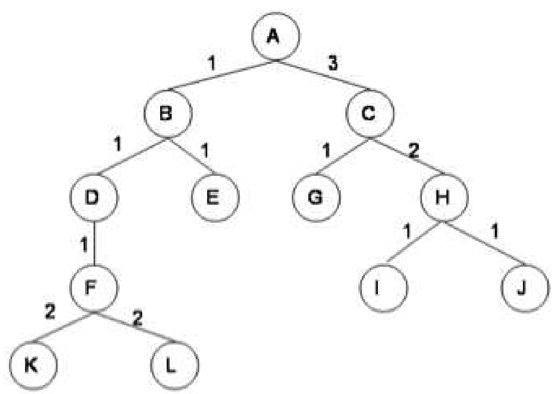
This mock exam is entirely voluntary. You *can* hand in solutions to your advisors if you’re unsure and you are free to ask questions about them. Solutions will be provided in the following week.

These exercises are all representative of actual exam questions.

1 Search

Problem 1.1 8 pt

Explain how BFS and DFS work and write down the sequences of nodes expanded for these algorithms. 4 min



Solution:

- BFS: ABCDEGHFIJKL
- DFS: ABDFKLECGHIJ

Problem 1.2 (Admissibility limits) 6 pt

The condition for a heuristic $h(n)$ to be admissible is that for all nodes n holds that $(0 \leq h(n) \leq h^*(n))$, where $h^*(n)$ is the true cost from n to goal. What happens when for all nodes, $h(n) = 0$ and when $h(n) = h^*(n)$? 3 min

---

Solution: When $h(n) = 0$, the search will behave like an uninformed search, and when $h(n) = h^*(n)$ the search will only expand the nodes on the optimal path to a goal.

Problem 1.3 6 pt

Does a finite state space always lead to a finite search tree? How about a finite space state that is a tree? Justify your answers. 3 min

Solution: No (there can be cycles). Yes if it's a tree (no cycles).

2 Adversarial Search

Problem 2.1 (Minimax Restrictions) 10 pt

Name at least five criteria that a game has to satisfy in order for the minimax algorithm to be applicable. 5 min

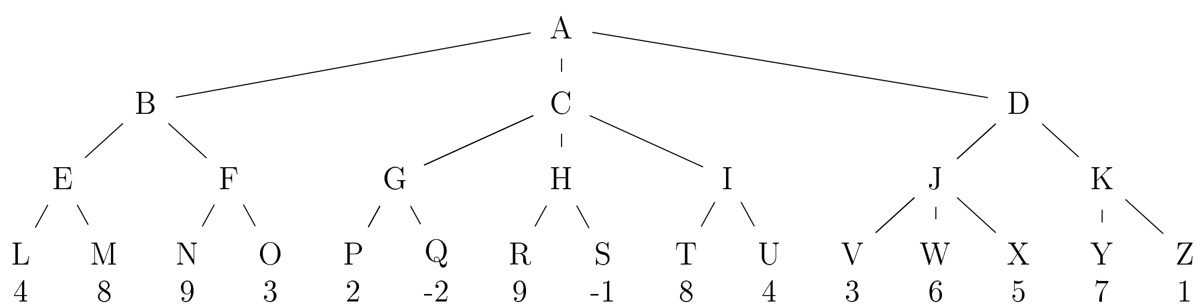
---

Solution: Any five of:

- Two-player
- Determininstic
- Fully observable
- Players alternate
- Finitely many / discrete game states
- Zero-sum
- Game ends after finitely many rounds

Problem 2.2 (Game Tree) 10 pt

Consider the following game tree. Assume it is the maximizing player's turn to move. The values at the leaves are the static evaluation function values of the states at each of those nodes. 5 min



20 pt

1. Label each non-leaf node with its minimax value. See above 5 pt
2. Which move would be selected by Max? 15 pt
3. List the nodes that the alpha-beta algorithm would prune (i.e., not visit). Assume children of a node are visited left-to-right. 10 pt
4. In general (i.e., not just for the tree shown above), if we traverse a game tree by visiting children in right-to-left order instead of left-to-right, can this result in a change to
   (a) the minimax value computed at the root?

(b) The number of nodes pruned by the alpha-beta algorithm?

Solution:

1. A:8, B:8, C:2, D:6, E:8, F:9, G:2, H:9, I:8, J:6, K:7
2. B
3. OHRSITUKYZ
4. (a) no, (b) yes

---

3 Constraint Satisfaction Problems & Inference

Problem 3.1 (CSP Heuristics) 14 pt

Explain backtracking search for CSPs and the minimum remaining values (MRV) heuristic, the degree heuristic and least constraining value heuristic (LCV). 7 min

Solution: Backtracking search is a search strategy for CSPs where we use DFS to assign one variable at a time and backtrack on finding inconsistent assignments. Backtracking follows inconsistent assignments to the original point of choice and adapts by selecting another possible value.

The MRV, degree and LCV are heuristics for (informed) backtracking search. MRV chooses the variable with the fewest possible assignments. The degree heuristic chooses the most constraining variable. LCV chooses the value which rules out the fewest values in the other variables.

Problem 3.2 (Scheduling CS Classes) 10 pt

You are in charge of scheduling for computer science classes that meet Mondays, Wednesdays and Fridays. There are 5 classes that meet on these days and 3 professors who will be teaching these classes. You are constrained by the fact that each professor can only teach one class at a time. The classes are: 15 min

- Class 1 - *Intro to Artificial Intelligence*: meets 8:30-9:30am,
- Class 2 - *Intro to Programming*: meets 8:00-9:00am,
- Class 3 - *Natural Language Processing*: meets 9:00-10:00am,
- Class 4 - *Machine Learning*: meets 9:30-10:30am,
- Class 5 - *Computer Vision*: meets 9:00-10:00am.

The professors are:

- Professor A, who is available to teach Classes 1, 2, 3, 4, 5.
- Professor B, who is available to teach Classes 3 and 4.
- Professor C, who is available to teach Classes 2, 3, 4, and 5.

1. Formulate this problem as a CSP problem in which there is one variable per class, stating the domains, and constraints. Constraints should be specified formally and precisely, but may be implicit rather than explicit. 3 pt
2. Give the constraint graph associated with your CSP. 2 pt
3. Show the domains of the variables after running arc-consistency on this initial graph (after having already enforced any unary constraints). 3 pt
4. List all optimal cutsets for the constraint graph associated with the CSP. 2 pt

Solution:

1.

Variables	Domains
C_1	A
C_2	A,C
C_3	A,B,C
C_4	A,B,C
C_5	A,C

Constraints: $C_1 \neq C_2$, $C_1 \neq C_3$, $C_1 \neq C_5$, $C_3 \neq C_4$, $C_3 \neq C_5$, $C_4 \neq C_5$

2.

$$\begin{array}{l} C_2 \\ \;| \\ C_1 - C_3 \\ \;\;\backslash \; / \; \backslash \\ \;\;\; C_5 - C_4 \end{array}$$

3.

Variable	Domain
C_1	A
C_2	C
C_3	B
C_4	A
C_5	C

4. The two optimal cutsets are $\{C_3\}$ and $\{C_5\}$.

---

4 Logic

Problem 4.1 (Calculus Properties) 8 pt 4 min

Explain briefly what the following properties of calculi mean:

- correctness
- completeness

Solution:

- correctness ($\mathcal{H} \vdash \mathbf{B}$ implies $\mathcal{H} \models \mathbf{B}$) - A calculus is correct if any derivable(provable) formula is also a valid formula.
- completeness ($\mathcal{H} \models \mathbf{B}$ implies $\mathcal{H} \vdash \mathbf{B}$) - A calculus is complete if any valid formula can also be derived(proven).

Problem 4.2 (An incorrect calculus) 6 pt 3 min

Why is this calculus $\mathcal{C}^2$ incorrect?

- $\mathcal{C}^2$ Axiom: $\mathbf{A} \vee \neg \mathbf{A}$
- $\mathcal{C}^2$ Inference Rules: $\dfrac{\begin{array}{c}[\mathbf{A}]^1 \\ \hline\hline \mathbf{B}\end{array}}{\mathbf{A} \Rightarrow \mathbf{B}} \Rightarrow I^1 \qquad \dfrac{\mathbf{A} \Rightarrow \mathbf{B} \quad \mathbf{B} \Rightarrow \mathbf{C}}{\mathbf{C}} \Rightarrow E$

Solution: In a correct calculus the inference rules must preserve validity. However the inference rule R2 does not preserve validity: Let $\mathbf{A} \Rightarrow \mathbf{B}$ and $\mathbf{B} \Rightarrow \mathbf{C}$ be valid. This is the case for an assignment for which $[\![\mathbf{A}]\!]_\varphi = \mathsf{F}$, $[\![\mathbf{B}]\!]_\varphi = \mathsf{F}$, and $[\![\mathbf{C}]\!]_\varphi = \mathsf{F}$. In particular the conclusion for this assignment is false: $[\![\mathbf{C}]\!]_\varphi = \mathsf{F}$. Hence the inference rule R2 does not preserve validity.

Problem 4.3 (Resolution Calculus) 6 pt

Use the resolution calculus to prove the validity of the expression: 3 min

$$(X \wedge Y) \vee (X \wedge \neg Y) \Rightarrow X$$

Solution:

$$(((X \wedge Y) \vee (X \wedge \neg Y)) \implies X)^F$$
$$((X \wedge Y) \vee (X \wedge \neg Y))^T; X^F$$
$$(X \wedge Y)^T \vee (X \wedge \neg Y)^T; X^F$$
$$X^T \vee (X \wedge \neg Y)^T; Y^T \vee (X \wedge \neg Y)^T; X^F$$
$$X^T; X^T \vee \neg Y^T; Y^T \vee X^T; Y^T \vee \neg Y^T; X^F$$
$$X^T; X^T \vee Y^F; Y^T \vee X^T; Y^T \vee Y^F; X^F$$

We get the clauses: $\{X^T\}, \{X^T, Y^F\}, \{Y^T, X^T\}, \{X^F\}$ and resolve: $\{X^T\}$ with $\{X^F\} \to \square$.

Problem 4.4 6 pt

Find three models for the following proposition: 3 min

$$(P \vee (Q \wedge R)) \Rightarrow (P \vee Q) \wedge (\neg P \vee R)$$

Solution: A valid model has to either satisfy $(P \vee Q) \wedge (\neg P \vee R)$ or not satisfy $(P \vee Q \wedge R)$. The latter is satisfied by e.g.

$$P \mapsto 0, Q \mapsto 0, R \mapsto 1$$
$$P \mapsto 0, Q \mapsto 1, R \mapsto 0$$
$$P \mapsto 0, Q \mapsto 0, R \mapsto 0$$
