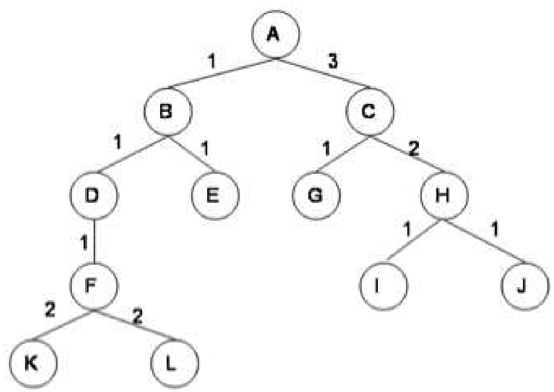
This mock exam is entirely voluntary. You *can* hand in solutions to your advisors if you're unsure and you are free to ask questions about them. Solutions will be provided in the following week.

These exercises are all representative of actual exam questions.

1 Search

Problem 1.1 Explain how BFS and DFS work and write down the sequences of nodes expanded for these algorithms. 8pt 4min



cstancumara

Problem 1.2 (Admissibility limits)

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 $h(n) = h^*(n)$? 6pt 3min

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.

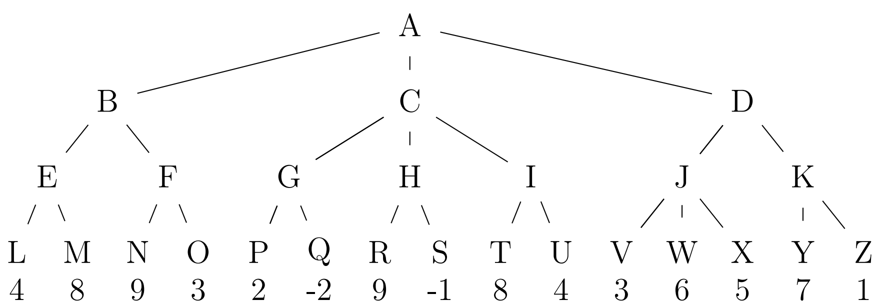
Problem 1.3 Does a finite state space always lead to a finite search tree? How about a finite space state that is a tree? Justify your answers. 6pt 3min

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

2 Adversarial Search

Problem 2.1 (Minimax Restrictions)
Name at least five criteria that a game has to satisfy in order for the minimax algorithm to be applicable. 10pt 5min

Problem 2.2 (Game Tree)
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. 10pt 5min



1. Label each non-leaf node with its minimax value. See above 10 pt
2. Which move would be selected by Max? 5 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 5 pt
   (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)

Explain backtracking search for CSPs and the minimum remaining values (MRV) heuristic, the degree heuristic and least constraining value heuristic (LCV). 14pt 7min

Problem 3.2 (Scheduling CS Classes)

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: 15pt 15min

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

The professors are:

- Professor A, who is available to teach Classes 3 and 4.
- Professor B, who is available to teach Classes 2, 3, 4, and 5.
- Professor C, who is available to teach Classes 1, 2, 3, 4, 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. 4 pt
2. Give the constraint graph associated with your CSP (e.g. by giving the edges). 2 pt
3. Show the domains of the variables after running arc-consistency on this initial graph (after having already enforced any unary constraints). 4 pt
4. Give one solution to this CSP. 1 pt
5. Your CSP should look nearly tree-structured. Briefly explain (one sentence or less) why we might prefer to solve tree-structures CSPs. 2 pt
6. Name (or briefly describe) a standard technique for turning these kinds of nearly tree-structured problems into tree-structured ones. 2 pt

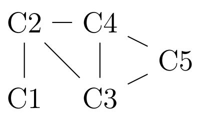
---

Solution:

1.

Variables	Domains
C1	C
C2	B,C
C3	A,B,C
C4	A,B,C
C5	B,C

Constraints: $C1 \neq C2$, $C2 \neq C3$, $C3 \neq C4$, $C4 \neq C5$, $C2 \neq C4$, $C3 \neq C5$

2. 

3.

Variable	Domain
C1	C
C2	B
C3	A,C
C4	A,C
C5	B,C

Note that C5 cannot possibly be C, but arc consistency does not rule it out.

4. C1 = C, C2 = B, C3 = C, C4 = A, C5 = B. One other solution is possible (where C3 and C4 are switched).

5. Minimal answer: Can solve them in polynomial time. If a graph is tree structured (i.e. has no loops), then the CSP can be solved in O(nd2) time as compared to general CSPs, where worst-case time is O(dn). For tree-structured CSPs you can choose an ordering such that every nodeâĂŹs parent precedes it in the ordering. Then you can greedily assign the nodes in order and will find a consistent assignment without backtracking.

6. Minimal answer: cutset conditioning. One standard technique is to instantiate cutset, a variable (or set of variables) whose removal turns the problem into a tree structured CSP. To instantiate the cutset you set its values in each possible way, prune neighbors, then solve the reduced tree structured problem (which is fast).

---

4 Logic

Note: For ASCII submissions, use the symbols `&`, |, `~`, `->` instead of $\wedge, \vee, \neg, \Rightarrow$.

dasenovlucia

Problem 4.1 (Calculus Properties)

Explain briefly what the following properties of calculi mean: 8pt

- correctness
- completeness

4min

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)

Why is this calculus $\mathcal{C}^2$ incorrect? 6pt

- $\mathcal{C}^2$ Axiom: $\mathbf{A} \vee \neg\, \mathbf{A}$

3min

- $\mathcal{C}^2$ Inference Rules: $\dfrac{\dfrac{[\mathbf{A}]^1}{\mathbf{B}}}{\mathbf{A} \Rightarrow \mathbf{B}} \Rightarrow I^1$ $\qquad \dfrac{\mathbf{A} \Rightarrow \mathbf{B} \quad \mathbf{B} \Rightarrow \mathbf{C}}{\mathbf{C}} \Rightarrow E$

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 $[\![\mathbf{A}]\!]_\varphi = \mathsf{F}$, $[\![\mathbf{C}]\!]_\varphi = \mathsf{F}$, 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)

Use the resolution calculus to prove the validity of the expression: 6pt

3min

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

Solution: We use the following tableau derivations:

$$\begin{array}{c} (X \wedge Y) \vee (X \wedge \neg Y) \Rightarrow X^{\mathsf{F}} \\ (X \wedge Y) \vee (X \wedge \neg Y)^{\mathsf{T}} \\ X^{\mathsf{F}} \\ \begin{array}{c|c} X \wedge Y^{\mathsf{T}} & X \wedge \neg Y^{\mathsf{T}} \\ X^{\mathsf{T}} & X^{\mathsf{T}} \\ Y^{\mathsf{T}} & \neg Y^{\mathsf{T}} \\ \bot & \bot \end{array} \end{array}$$

Therefore there is no model. Thus the initial expression is unsatisfiable.

Therefore the expression

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

is valid.

Problem 4.4 Find three models for the following proposition: 6pt

3min

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

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.

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

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

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