
Midterm: Th. Nov.03, 2pm - 3pm

Intro AI ICS 171
Instructor: Max Welling

- *This exam is closed book*
- *Spend your time wisely: get a shot at each question.*
- *You can get a total of 20 points.*
- *Good Luck !*

- 1.(2pts) **Chess Agents** Consider two intelligent agents playing chess with a clock. One of them is called “Deep Blue”, while the other is called Gary Kasparov.
- a.(1pt) Roughly specify the task environment for Deep Blue. (This means specify each letter in “PEAS”).
- a) answer: a) performance measure: winning the game, environment: chess-pieces on a chess-board, adversary, c) actuators: screen, d) camera, keyboard.
- b.(1pt) Determine each of the following properties of this task environment: a) fully observable or partially observable, b) deterministic or stochastic, c) episodic or sequential, d) static, dynamic, or semidynamic e) discrete or continuous, f) single agent or multi-agent. Explain your answer.
- b) answer: a) fully observable, b) deterministic or strategic are both ok. c) sequential, d) semidynamic, e) discrete f) multi-agent.
- 2.(5pts) **The 8-puzzle** Consider the 8-puzzle problem described in the book and homework.
- a.(1pt) We like to search for a solution using A^* -search. Describe the following aspects of the problem formulation: a) states, b) successor function, c) goal test, d) step cost, e) path cost.
- a) answer: a) a configuration of the 8 squares. b) all possible moves of the empty square, c) if end state has been reached, d) say 1 for each move, e) the total number of moves up till the current state.
- b.(2pts) For A^* -search the evaluation function, $f(n)$, consists of the path cost, $g(n)$, plus the heuristic function, $h(n)$. We will use the Manhattan distance between the current state and the goal state to be the heuristic function. Explain when a heuristic function is admissible and prove this fact for the Manhattan distance heuristic.
- b) answer: A heuristic is admissible if it underestimates the true cost. Manhattan distance ignores the constraint that other squares may block its way. Relaxing constraints always underestimates the true cost.

- c.(2pts) Describe when a heuristic is consistent and proof this for the Manhattan distance heuristic.
- c) answer: A heuristic is consistent if the "triangle inequality" holds. Manhattan distance is the shortest path on a grid. Hence taking a detour will always increase the distance.

THERE ARE TWO MORE QUESTIONS ON THE NEXT PAGE!

- 3.(7pts) **Graph Coloring** Consider a graph with 4 nodes corresponding to variables numbered 1,2,3,4,5 and edges between the following nodes: 1-2, 2-3, 3-4, 4-5, 1-5, 1-4, 2-4, 3-5. Each variable can take three values: A, B, or C. Two variables corresponding to nodes which are connected by an edge must have different values.
- (1pt) What is the domain for each variable? Draw the constraint graph.
 - answer: $D=[A,B,C]$.
 - (2pts) Solve this problem by hand by using the "minimum remaining value" heuristic to choose the next node to expand in the search tree. Break ties using the "degree heuristic". Use the "least-constraining-value" heuristic to pick a value. Any remaining ties can be broken arbitrarily. At each step, explain your reasoning.
 - answer: 4-A, 1-B, 2-C, 3-B, 5-C. There are many solutions because many ties need to be broken arbitrarily.
 - (2pts) Assume that node 1 has value C and node 5 has value B. Apply forward checking to reduce the domains of (some) of the other variables. Explain your answer.
 - answer: $D2=[A,B]$, $D3=[A,C]$, $D4=[A]$.
 - (2pts) Use arc-consistency (repeatedly) to further reduce the domains of the remaining variables. Explain your answer.
 - answer: $D3=[C]$, $D2=[B]$.
- 4.(6pts) **Games** Consider the following 3-ply game: First MAX moves and has 2 choices (L or R), then MIN moves and has also 2 choices (L or R), finally MAX moves again and has two choices (L or R). So the total number of possible games add up to 8. The pay-offs for each possible game for MAX are as follows: RRR=6, RRL=5, RLR=8, RLL=7, LRR=2, LRL=1, LLR=4, LLL=3. For example: if MAX moves R, MIN moves R and MAX moves R again will pay 6 to MAX. MAX tries to maximize it's pay-off and MIN tries to minimize MAX's pay-off.
- (2pts) Draw the 3-ply game tree for this game including the usual leaf-nodes which contain the pay-off values for MAX. Assign to every node in the tree the best pay-off for MAX in that branch. For instance, the root node should contain the value that is paid to MAX after actually playing the game rationally.
 - answer: Root=6,L=2,R=6,LL=4,LR=2,RR=6,RL=8.
 - (3pts) Search the game tree using the MINIMAX algorithm and alpha-beta pruning. R moves should be explored before L moves, when the MINIMAX algorithm has no preference. Which nodes need NOT be visited if we apply alpha-beta pruning? Each time you decide a branch can be pruned, redraw the part of the game tree that has been visited already and explain why the branch can be pruned. It is recommended that you maintain bounds on the possible values of each node while searching the tree, and explain each pruning step using those bounds (similarly to what was done in class).
 - answer: Branches: RLL, LLL and LLR can be pruned.
 - (1pt) What are the worst case time and space complexity of the MINIMAX search algorithm?
 - answer: space: $O(bm)$ or $O(m)$, time $O(b^m)$ with b the branching factor and m the depth of the solution.