CS 188 Fall 2015 — Introduction to Artificial Intelligence — Midterm

- You have approximately 80 minutes.
- The exam is closed book, closed calculator, and closed notes except your one-page crib sheet.
- Mark your answers ON THE EXAM ITSELF. If you are not sure of your answer you may wish to provide a *brief* explanation. All short answer sections can be successfully answered in a few sentences AT MOST.

First name	
Last name	
SID	
edX username	

Name of person on your left	
Name of person on your right	

For staff use only:

Q1.	True/False and Multiple Choice	/20
Q2.	Search Problems	/30
Q3.	Heuristics	/15
Q4.	Fair Play	/15
Q5.	Propositional Logic	/20
	Total	/100

Q1. [20 pts] True/False and Multiple Choice

(a) For each True/False question, circle the correct answer. Missing choices and wrong choices with no explanation are worth zero.

(i) [2 pts] [*true* or *false*] Given any tree, it's always possible to design an admissible heuristic which makes A* search equivalent to uniform cost search.

(ii) [2 pts] [*true* or *false*] Two different search algorithms may give different results on the same constraint satisfaction problem.

(iii) [2 pts] [*true* or *false*] Improving an agent's initial observation and providing it with a more accurate transition model will cause it to have larger belief states.

(iv) [2 pts] [*true* or *false*] A CSP can only have unary and binary constraints.

(v) [2 pts] [*true* or *false*] Iterative deepening search needs more memory than breadth-first search.

(vi) [2 pts] [*true* or *false*] No logical agent can behave rationally in a partially observable environment.

(b) Agents and Environments

Select the **better** example of a task environment for each of the following attributes.

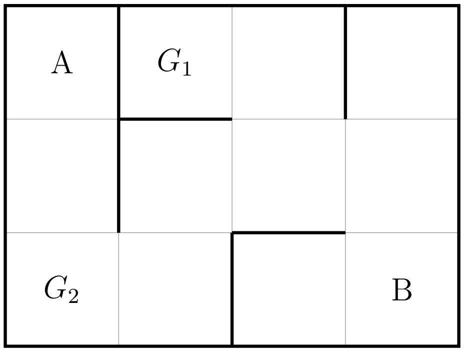
(i) [2 pts] **Partially Observable**
○ Image Processing System ○ Medical Diagnosis System

(ii) [2 pts] **Continuous**
○ Self-Driving Car ○ Pacman

(iii) [2 pts] **Stochastic**
○ Backgammon ○ Sudoku

(iv) [2 pts] **Static**
○ Chat Room Agent ○ Checkers

Q2. [30 pts] Search Problems



Alice (A) and Bob (B) are in a maze of height M and width N. They know the layout of the maze, including the size, locations of all walls, and locations of the K goal squares $G_1, \ldots, G_K$. K is between 0 and MN, and is a known quantity. Alice and Bob know their own location, as well as the other person's location, at all times. Their goal is to meet up at the *same* goal square in as few time steps as possible.

At each time step, Alice and Bob act simultaneously, and they can both either move to an adjacent square, **or stay put (stop)**. Alice starts in the top left corner of the maze, and Bob starts in the bottom right corner.

(a) [4 pts] Give the state representation for the minimal state space for this problem (i.e., do not include extra information). You should answer for a **general instance of the problem**, not the specific maze shown.

(b) [2 pts] What is the size of this state space?

(c) [3 pts] What is the maximum branching factor for this search problem? Remember that stay put (stop) is also a choice of actions for Alice and Bob.

(d) [6 pts] Suppose we use breadth-first graph search to solve this problem. If we now consider the maze above, of size 3 by 4, which solution will BFS produce? Specifically, which goal square will they end up at, and how many time steps would it take for both Alice and Bob to reach that goal square?

Goal state:

Time steps:

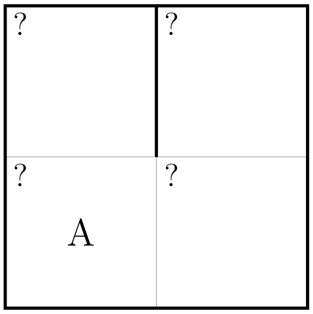
Now suppose that the locations of the K goal squares $G_1, \ldots, G_K$ are unknown, and K is also unknown, though it is still known to be between 0 and MN. When either Alice or Bob enters a square, he/she can determine if it is a goal square, and because the two can communicate with each other, both share that information.

(e) [4 pts] What is the size of the initial belief state, before any percepts are received (i.e., Alice and Bob don't know if they are on a goal)?

(f) [3 pts] After receiving the initial percept, but before they take any actions, Alice and Bob both determine that their starting positions are not goal squares. What is the size of the new belief state?

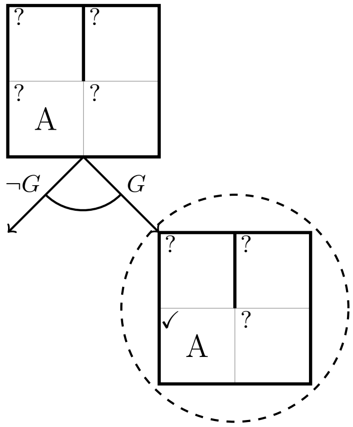
(g) [3 pts] If, instead of K being unknown, Alice and Bob received prior knowledge that K was *exactly* one, what would be the size of the initial belief state, before any percepts are received?

Let's now consider a simplified setting, where Alice is alone in a smaller maze. The locations of the goal squares are still unknown, and Alice has to find a goal square. **Staying in the same square is no longer a valid action.** Assume that this is our starting state, before Alice receives any percepts:



(h) [5 pts] Below is an incomplete And-Or search tree for this new problem. The first layer is complete, and the second layer is partially done. Fill in the missing parts **up to and including the fourth layer of the tree**.

- For the nodes of the tree, draw out the board configurations. A complete board configuration will include the position of Alice, and a "✓", "?", or "X" for every square, denoting whether that square is known to be a goal, still unknown, or known not to be a goal, respectively.
- Label the edges one of [G, $\neg G$, Up, $Down$, $Left$, $Right$] depending on the edge type, and the percept received or the action taken. G indicates a Goal percept and $\neg G$ indicates a No-goal percept.
- Mark AND with an arc across the branches, as shown in the beginning of the tree below. Conversely, you will indicate OR by **not** including an arc across the branches.
- Circle all terminal nodes. One has been circled for you.
- Hint: there will be nine total boxes, including the two we gave you.



Q3. [15 pts] Heuristics

For parts a, b, c below, consider now the CornersProblem from Project 1: there is a food pellet located at each corner, and Pacman must navigate the maze to find each one.

(a) [5 pts] For each of the following heuristics, say whether or not it is admissible. If a heuristic is inadmissible, give a concrete counterexample (i.e., draw a maze configuration in which the value of the heuristic exceeds the true cost).

- h_1 is the maze distance to the nearest food pellet (if no food pellets remain, $h_1 = 0$).

  [*Admissible* or *Not-Admissible*]

- h_2 is the number of uneaten food pellets remaining.

  [*Admissible* or *Not-Admissible*]

- $h_3 = h_1 + h_2$

  [*Admissible* or *Not-Admissible*]

- $h_4 = |h_1 - h_2|$

  [*Admissible* or *Not-Admissible*]

- $h_5 = \max\{h_1, h_2\}$

  [*Admissible* or *Not-Admissible*]

(b) [2 pts] Pick one heuristic from part (a) that you said was inadmissible; call it h_k. Give the smallest constant $\epsilon > 0$ such that $h' = h_k - \epsilon$ is an admissible heuristic. Briefly justify your answer.

(c) [3 pts] We will say that for two heuristics h_a and h_b, $h_a \leq h_b$ if for all possible states x, $h_a(x) \leq h_b(x)$. In this case, we say h_b *dominates* h_a. Fill in the boxes below with all of the heuristics that you said were admissible in part (a) so that all heuristics are to the right of any heuristics they dominate. If two heuristics share no dominance relationship, put them in the same box. You may not need to use all the boxes.

□ ≤ □ ≤ □ ≤ □ ≤ □

(d) [2 pts] Let h_i and h_j be two admissible heuristics and let $h_\alpha = \alpha h_i + (1 - \alpha) h_j$. Give the range of values for α for which h_α is guaranteed to be admissible.

(e) [3 pts] Consider an arbitrary search problem in a graph with start state S and goal state G. Let h^* be the "perfect heuristic," so $h^*(x)$ is the optimal distance from from x to the goal G. Let ϵ be some positive number.

For each of the heuristics h_A, h_B, and h_C, give the range of possible values for the cost of a path that A* tree search could return when using that heuristic. You may write your answers in terms of $h^*(S)$, and ϵ.

Let h_0 be an arbitrary admissible heuristic.

(i) [1 pt] h_A: $h_A = h_0$ for all states except at one unspecified state y, where $h_A(y) = h_0(y) + \epsilon$

(ii) [1 pt] h_B: $h_B(x) = h_0(x) + \epsilon$ for all states x.

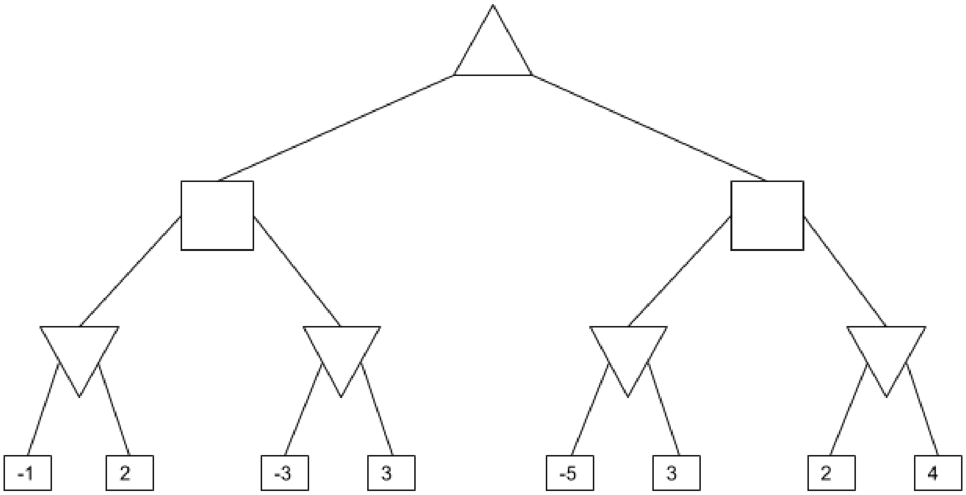
(iii) [1 pt] h_C: $h_C(x) = h_0(x) + \epsilon$ for all states x except the goal state G, where $h_C(G) = h_0(G) = 0$.

Q4. [15 pts] Fair Play

Consider a game tree with three agents: a maximizer, a minimizer, and an equalizer. The maximizer chooses the highest score, the minimizer chooses the lowest score, and the equalizer chooses tries to *minimize the absolute value* (i.e. equalizer wants to make the game as close as possible, so it chooses whichever value is closest to zero).

We use an upward-facing triangle to represent a max node, a downward-facing triangle to represent a min node, and a square to represent an equalizer node. the values in the leaves are given from max's point of view.

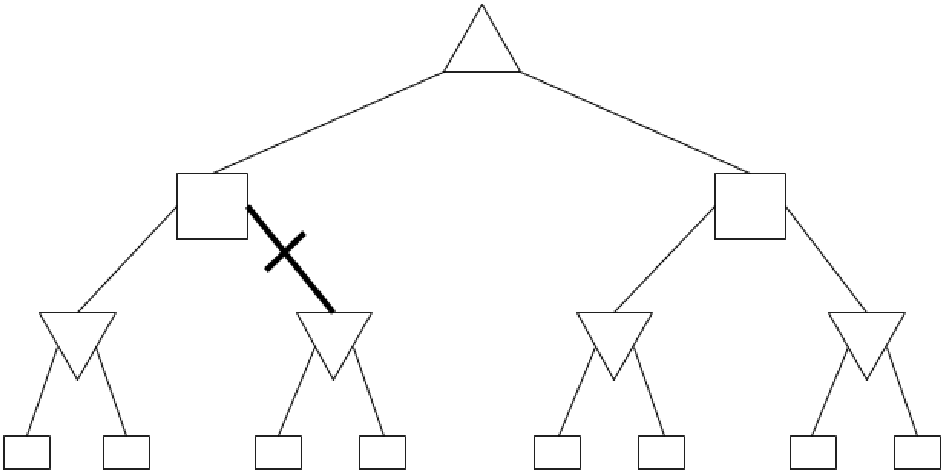
(a) [2 pts] Fill in all values in the game tree below:



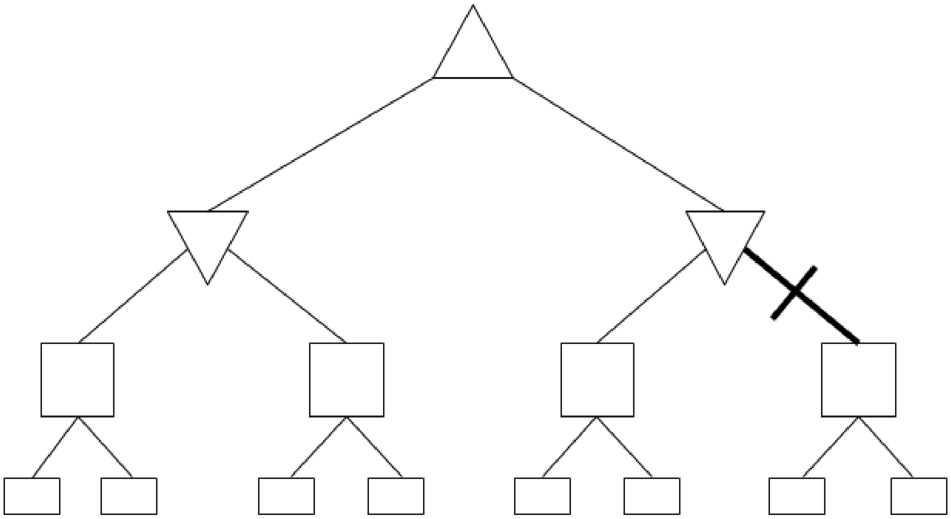
(b) [5 pts] In the same game tree above, put an X on the line of all branches that can be pruned, or write "No pruning possible." Assume that branches are explored from left to right.

(c) [8 pts] For each of the following game trees, fill in values in the leaf nodes such that only the marked, bold branches can be pruned. Assume that branches are explored from left to right. If no values will allow the indicated nodes to be pruned, write "Not possible." **Be very clear**: if you write "Not possible," we will not look at the values you filled in.

(i) [4 pts] [Hint: what is the *best possible value* from the equalizer's viewpoint?]



(ii) [4 pts] Note that the order of the players has changed in the game tree below.



Q5. [20 pts] Propositional Logic

(a) [4 pts] Pacman has lost the meanings for the symbols in his knowledge base! Luckily he still has the list of sentences in the KB and the English description he used to create his KB.

For each English sentence on the left, there is a corresponding logical sentence in the knowledge base on the right (**not necessarily the one across from it**). Your task is to recover this matching. Once you have, please fill in the blanks with the English sentence that matches each symbol.

English
There is a ghost at (0, 1).
If Pacman is at (0, 1) and there is a ghost at (0, 1), then Pacman is not alive.
Pacman is at (0, 0) and there is no ghost at (0, 1).
Pacman is at (0, 0) or (0, 1), but not both.

Knowledge Base
$(C \vee B) \wedge (\neg C \vee \neg B)$
$C \wedge \neg D$
$\neg A \vee \neg(B \wedge D)$
D

$A =$______________________________ $B =$______________________________

$C =$______________________________ $D =$______________________________

(b) [8 pts] Consider a generic propositional model with 4 symbols: A, B, C, D. For the following sentences, mark how many models containing these 4 symbols will satisfy it.

(i) [2 pts] $\alpha_1 = A$: ______________________________

(ii) [3 pts] $\alpha_2 = (C \wedge D) \vee (A \wedge B)$: ______________________________

(iii) [3 pts] $\alpha_3 = (A \vee B) \Rightarrow C$: ______________________________

(c) The DPLL satisfiability algorithm is a backtracking search algorithm with 3 improvements: PURE-SYMBOLS, UNIT-CLAUSES, and EARLY TERMINATION. In this question, we'll ask you to relate these improvements (if possible) to more general CSP techniques used with backtracking search.

(i) [2 pts] The PURE-SYMBOL technique finds a propositional symbol that only occurs with the same "sign" throughout the expression and assigns it the corresponding value. Such symbols are called pure. In the following CNF expression, which symbols are pure and what value does this process assign to those symbols?

$$(C \vee D) \wedge (C \vee \neg A) \wedge (\neg D \vee A) \wedge (\neg B \vee A)$$

pure symbols(s): ______________________________

value(s) assigned: ______________________________

(ii) [2 pts] Which of the following CSP techniques is equivalent to the PURE-SYMBOLS technique when applied to SAT for CNF sentences:

○ MINIMUM-REMAINING-VALUES
○ FORWARD-CHECKING
○ LEAST-CONSTRAINING-VALUE
○ BACKTRACKING
○ No equivalent CSP technique

(iii) [2 pts] The UNIT-CLAUSE technique finds all clauses that contain a single literal and assigns values to those literals. Which of the following CSP techniques is equivalent to the UNIT-CLAUSE technique when applied to SAT for CNF sentences:

○ MINIMUM-REMAINING-VALUES
○ FORWARD-CHECKING
○ LEAST-CONSTRAINING-VALUE
○ BACKTRACKING
○ No equivalent CSP technique

(iv) [2 pts] DPLL performs early termination in two steps: SUCCESS-DETECTION and FAILURE-DETECTION. First, SUCCESS-DETECTION checks to see if *all* clauses evaluate to true. If so, the sentence is satisfiable and any unassigned propositional symbols can be assigned arbitrarily. Next, FAILURE-DETECTION checks if *any* clause evaluates to false. In this case the sentence is unsatisfiable and this branch of the search can be pruned. How does this strategy relate to the early termination strategy used by the general BACKTRACKING algorithm?

○ BACKTRACKING does both SUCCESS-DETECTION and FAILURE-DETECTION
○ BACKTRACKING does only SUCCESS-DETECTION
○ BACKTRACKING does only FAILURE-DETECTION
○ BACKTRACKING does neither