15-381/781: Artificial Intelligence **Name:** ____________________

Fall 2016

Practice Midterm

80 minutes

This is a practice midterm examination. It was designed to be completed in 80 minutes. **The points assigned to the questions are arbitrary and should be ignored.**

---

1. (10 points) Consider a heuristic search instance. For any state x, let $h^*(x)$ be the shortest distance between x and goal state t. Let $h(\cdot)$ be a heuristic that overestimates $h^*(\cdot)$ by *exactly* a constant $c > 0$, that is for any state x, $h(x) = h^*(x) + c$.

   Assume that the search instance has a unique optimal path between start state s and goal node t. Show that A* *tree search* only expands the nodes that are on the optimal s-t path.

2. (10 points) Consider a robot that is moving in an environment. The goal of the robot is to move from an initial point to a destination point as fast as possible. However, the robot has the limitation that if it moves fast, its engine can overheat and stop the robot from moving. The robot can move with two different speeds: slow and fast. If it moves fast, it gets a reward of 10; if it moves slowly, it gets a reward of 4. We can model this problem as an MDP by having three states: cool, warm, and off. The transitions are shown below. Assume that the discount factor is $\gamma = 0.9999$ and also assume that when we reach the state off, we remain there without getting any reward.

s	a	s'	$T(s, a, s')$
cool	slow	cool	1
cool	fast	cool	1/2
cool	fast	warm	1/2
warm	slow	cool	1/2
warm	slow	warm	1/2
warm	fast	warm	1/2
warm	fast	off	1/2

1. Consider the conservative policy π where the robot always moves slowly. What is the value of $V^\pi(\text{cool})$? Remember that $V^\pi(s)$ is the expected discounted sum of future rewards when starting at state s and following policy π.

2. What is the optimal policy for each state? Justify your answer.

3. Is it possible to change the discount factor to get a different optimal policy? If yes, give such a change and the new optimal policy. If no justify your answer in at most two sentences.

3. (10 points) Consider an arbitrary MDP M_1 with states S, transition model $T(s, a, s')$, reward model $R(s, a, s')$ and discount factor γ. Now define MDP M_2 as exactly the same MDP, except that its reward model is $R(s, a, s')+d$, where d is a constant (in other words, all the rewards in M_2 are exactly the same as in M_1, except with an additional d added on). Prove that after running to convergence over an infinite horizon, M_2 has exactly the same optimal policy (or optimal policies) as M_1. You can assume that there are no terminal states in M_1 and M_2.

4. (5 points) *For this question, recall from class that a heuristic $h(\cdot)$ is said to dominate heuristic $g(\cdot)$, if for all states x, $h(x) \geq g(x)$.*

   Let $h_1(\cdot)$ and $h_2(\cdot)$ be two arbitrary admissible heuristics. Let $h_3 = \frac{2}{3}h_1 + \frac{1}{3}h_2$.

   1. Prove or disprove: h_3 is an admissible heuristic.

   2. Prove or disprove: h_3 dominates one or both of h_1 and h_2.

5. (5 points) A patient has tested positive for the disease x. Let pos/neg indicate testing positive or negative for the disease respectively, and let $x/\neg x$ indicate having disease x or not. You are given the following data:

   - If a person has the disease, the probability of testing positive is $p(pos|x) = 0.99$.
   - If a person does not have the disease, the probability of testing negative is $p(neg|\neg x) = 0.99$.
   - One out of ten thousand people have x, so the probability of having x prior to testing is $p(x) = 0.001$.

   Write the formula for computing $p(x|pos)$, the probability of having the disease x given that the patient tested positive. You do not have to calculate the final value, but your answer should be in terms of raw numbers and mathematical operators like plus, minus, times and division, so that we could immediately evaluate it using a calculator.

6. (6 points) Prove or disprove: Consider a STRIPS planning problem where every operation has no preconditions, and for every condition there is some operation that has it as a postcondition. Then in this problem's planning graph, all of the conditions will appear in level S_1 (the second level of conditions), and there are no mutex relations on that level except between pairs of conditions such that one is the negation of the other.