CS 188 Spring 2006

Introduction to Artificial Intelligence

# Practice Final

You have 180 minutes. The exam is open-book, open-notes, no electronics other than basic calculators. 100 points total. Don't panic!

Mark your answers ON THE EXAM ITSELF. Write your name, SID, login, and section number at the top of each page.

For true/false questions, CIRCLE *True* OR *False*.

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.

**1. (20 points.) True/False**

*Each problem is worth 2 points. Incorrect answers are worth 0 points. Skipped questions are worth 1 point.*

(a) *True/False*: All MDPs can be solved using expectimax search.

(b) *True/False*: There is some single Bayes' net structure over three variables which can represent any probability distribution over those variables.

(c) *True/False*: Any rational agent's preferences over outcomes can be summarized by a single real valued utility function over those outcomes.

(d) *True/False*: Temporal difference learning of optimal utility values (U) requires knowledge of the transition probability tables (T).

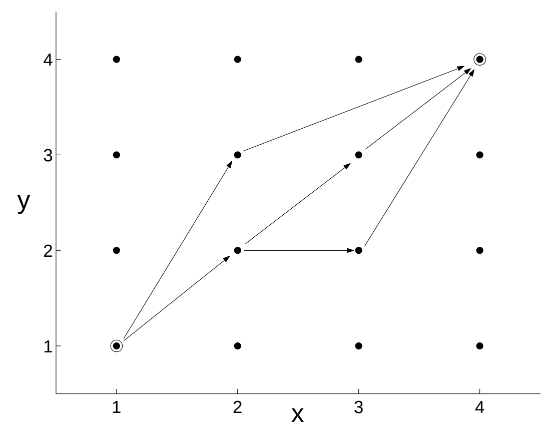
(e) *True/False*: Pruning nodes from a decision tree may have no effect on the resulting classifier.

**2. (24 points.) Search**

Consider the following search problem formulation:

**States**: 16 integer coordinates, $(x, y) \in [1, 4] \times [1, 4]$
**Initial state**: $(1, 1)$
**Successor function**: The successor function generates 2 states with different $y$-coordinates
**Goal test**: $(4, 4)$ is the only goal state
**Step cost**: The cost of going from one state to another is the Euclidean distance between the points

We can specify a state space by drawing a graph with directed edges from each state to its successors:




**Uninformed Search** Consider the performance of DFS, BFS, and UCS on the state space above. Order successors so that DFS or BFS explores the state with lower $y$-coordinate first.

**a)** What uninformed search algorithm(s) find an *optimal* solution? What is this path cost?

**b)** What uninformed search algorithm(s) find a *shortest* solution? How long is this path?

**c)** What uninformed search algorithm(s) are most *efficient*? How many search nodes are expanded?

**Heuristic Search** Use the Euclidean distance to the goal as a heuristic for $A^*$ and greedy best-first search:

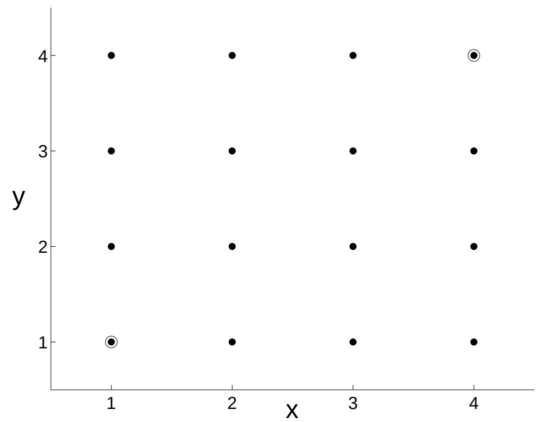
**d)** What heuristic search algorithm(s) find an *optimal* solution?

**e)** What heuristic search algorithm(s) find a *shortest* solution?

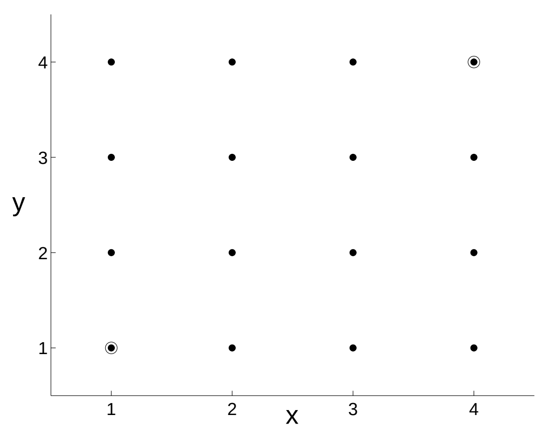
**f)** What heuristic search algorithm(s) are most *efficient*? How many search nodes are expanded?

In this family of grid-based problems, we can specify a state space by drawing a graph with directed edges from each state to its successors. Assume the search procedures order successors so that DFS or BFS explores the state with lower $y$-coordinate first.

**a)** Show a state space in which DFS and BFS are more efficient than $A^*$ with the Euclidean heuristic.




**b)** Is the Manhattan distance to the goal an admissible heuristic for this search problem? If so, prove it; otherwise, show a state space in which $A^*$ with the Manhattan heuristic is not optimal.



**3. (16 points.) CSPs**

[From a previous year's exam.] Consider the problem of tiling a surface (completely and exactly covering it) with $n$ dominoes (each is a $2 \times 1$ rectangle, and the number of pips is irrelevant). The surface is an arbitrary edge-connected (i.e., adjacent along an edge, not just a corner) collection of $2n$ $1 \times 1$ squares (e.g., a checkerboard, a checkerboard with some squares missing, a $10 \times 1$ row of squares, etc.).

**a)** Formulate this problem precisely as a CSP where the dominoes are the variables (i.e., define the variable domain and the constraints).

**Variables**: $X_1, \ldots, X_n$
**Domains**:

**Constraints**:

**b)** Formulate this problem precisely as a CSP where the squares are the variables (i.e., define the variable domain and the constraints). [*Hint: it doesn't matter which particular domino covers a given pair of squares.*]

**Variables**: $X_1, \ldots, X_{2n}$
**Domains**:

**Constraints**:

**c)** For your formulation in part (b), describe exactly the set of instances that have a tree-structured constraint graph.

**d)** [Extra credit] Consider domino-tiling a checkerboard in which two opposite corners have been removed (31 dominoes to tile 62 squares). Prove that the CSP has no consistent assignment.

**4. (20 points.) Probabilistic Reasoning**

Consider the problem of predicting a coin flip on the basis of several previous observations. Imagine that you observe three coin flips, and they all come up heads. You must now make a prediction about the (distribution of the) outcome of a fourth flip.

a) What is the maximum likelihood estimate for the probability that the fourth flip will be heads?

Assume that there are three kinds of coins in the world: fair coins, two-headed coins which always come up heads, and two-tailed coins which always come up tails. Moreover, assume that you have a prior belief or knowledge that a given coin is fair with probability 0.5, and two-headed or two-tailed with probability 0.25 each.

b) Draw a Bayes' net which expresses the assumptions that a single coin is chosen, then the flip outcomes are indentically distributed and conditionally independent given the coin type. Your network should include a node for the unobserved fourth flip.

c) What is the posterior probability that the coin is fair given the three heads observations?

d) What is the probability that the fourth flip will be heads given the three heads observations?

**5. (20 points.) Tree Search**

[Adapted from a previous exam] In the following, a "max" tree consists only of max nodes, whereas an "expectimax" tree consists of a max node at the root with alternating layers of chance and max nodes.

**(a)** Assuming that leaf values are finite but unbounded, is pruning (as in alpha-beta) ever possible in a max tree? Give an example, or explain why not.

**(b)** Is pruning ever possible in an expectimax tree under the same conditions? Give an example, or explain why not.

**(c)** If leaf values are constrained to be nonnegative, is pruning ever possible in a max tree? Give an example, or explain wh

**(d)** If leaf values are constrained to be nonnegative, is pruning ever possible in an expectimax tree? Give an example, or ex

**(e)** If leaf values are constrained to be in the range $[0, 1]$, is pruning ever possible in a max tree? Give an example, or explai

**(f)** If leaf values are constrained to be in the range $[0, 1]$, is pruning ever possible in an expectimax tree? Give an example (qualitatively different from your example in (e), if any), or explain why not.

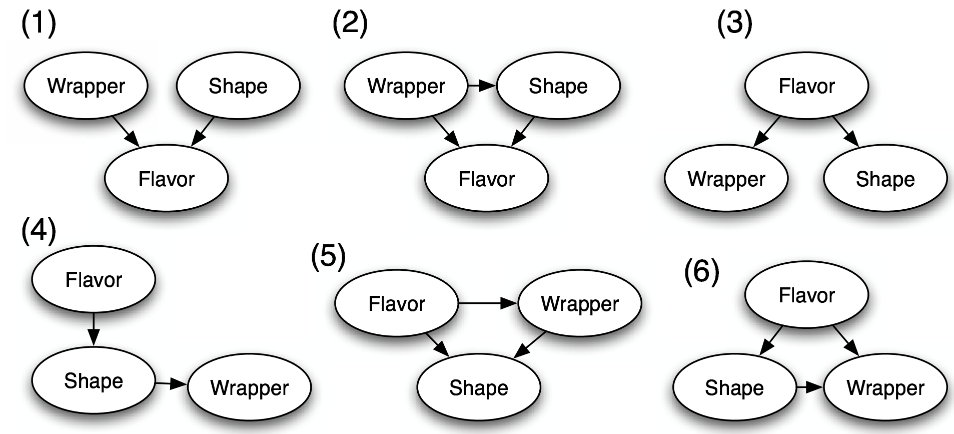
**(g)** Consider the outcomes of a chance node in an expectimax tree. Which of the following orders is most likely to yield pruning opportunities?

- Lowest probability first
- Highest probability first
- Doesn't make a difference

**6. (20 points.) Bayes' Nets**

[Adapted from a previous exam] The Surprise Candy Company makes candy in two flavors: 30% are Anchovy (which smell of the salty sea) and 70% are Berry (which smell delightful. All candies start out round and look the same. Someone (who can smell) trims some of the candies so that they are square. Then, a second person who *can't* smell wraps each candy in a red or brown wrapper. 80% of Berry candies are round and 74% have red wrappers. 90% of Anchovy candies are square and 82% have brown wrappers.

All candies are sold individually in sealed, identical, black boxes! You have just bought a box, but haven't opened it yet.




**(a)** Which network(s) can correctly represent $P(Flavor, Wrapper, Shape)$?

**(b)** Which network(s) assert(s) $P(Wrapper|Shape) = P(Wrapper)$?

**(c)** Which network(s) assert(s) $P(Wrapper|Shape, Flavor) = P(Wrapper|Shape)$?

**(d)** From the problem description, what independence relationships should hold for this problem? Which network is the best representation of this problem?

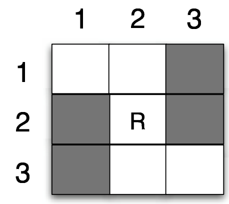
**(e)** What is the probability that the candy has a red wrapper?

**(f)** In the box is a round candy with a red wrapper. What is the probability that it is a Berry candy?

**(g)** If you tried to model the problem with network (2), can you give $P(Wrapper = Red|Shape = Round)$? If so, give the answer.

**(h)** If you tried to model the problem with network (4), can you give $P(Wrapper = Red|Shape = Round)$? If so, give the answer.

**7. (20 points.) HMMs**




**Robot Localization Grid**

Suppose you are a robot navigating a maze (see figure 1), where some of the cells are free and some are blocked. At each time step, you are occupying one of the free cells. You are equipped with sensors which give you noisy observations, $(w_U, w_D, w_L, w_R)$ of the four cells adjacent to your current position (*up,down,left*, and *right* respectively). Each $w_i$ is either *free* or *blocked*, and is accurate 80% of the time, independently of the other sensors or your current position. [1]

Imagine that you have experienced a motor malfunction that causes you to randomly move to one of the four adjacent cell with probability $\frac{1}{4}$.[2]

**a)** Suppose you start in the central cell in figure 1. One time step passes and you are now in a new, possibly different state and your sensors indicate (*free,blocked,blocked,blocked*). Which states have a non-zero probability of being your new position?

**b)** Give the posterior probability distribution over your new position.

**c)** Suppose that $s_0$ is your starting state and that $s_1$ and $s_2$ are random variables indicating your state after the first and second time steps. Draw a Bayes' net illustrating the relationships between each $s_i$ and the sensor observations associated with that state. Consider the CPT for $s_1$. How many values for $s_1$ have non-zero probability?

[1] Assume that if a cell is off the given grid it is treated as blocked.
[2] If you move towards a blocked cell, you hit the wall and stay where you are.

**d)** Suppose that $p_t(s)$ is a function representing the proability that at time step $t$, you are in state $s$. Using your bayes net diagram from (c), write a recursive function for $p_{t+1}(s)$ in terms of $p_t(s)$ and the observations $(w_U, w_D, w_L, w_R)$ for time step $t+1$. Make sure your formulation is general and not specific to the particular grid in figure 1.

**e)** Now suppose that your motor is not malfunctioning, but is instead being controlled by a person. You have no information about what your controller intends for your path to be, but you do know that your movements tend to continue in a given direction for a while before changing directions. More specifically, assume that at each time step, you will either be moved in the same direction as the previous times step (perhaps bouncing you into a wall) or that you will move in a random new direction (possibly resulting in the same direction again). Draw a bayes net representing this situation for three time steps i.e $s_0, s_1, s_2$. Make sure to briefly describe any new variables you introduce into the diagram. Indicate (in at most one sentence) how this preference for going in the same direction will be incoporated into the CPT of the nodes in your net.

**8. (20 points.) MDP Question**

[Adapted from Sutton and Barto] A cleaning robot must vacuum a house on battery power. It can at any time either *clean* the house, *wait* and do nothing, or *recharge* its battery. Unfortunately, it can only perceive its battery level (its state) as either *high* or *low*. If the robot recharges, the battery return to *high*, and the robot receives a reward of 0. If the robot waits, the battery level is unchanged, and the robot receives a reward of $R_{wait}$. If the robot cleans, the outcome will depend on the battery level. If the battery level is high, the battery drops to low with fixed probability 1/3. If the battery level is low, it runs out with probability 1/2. If the battery does not run out, the robot receives a reward of $R_{clean}$. If the battery does run out, a human must collect the robot and recharge it. In this case, the robot ends up with a high battery, but receives a reward of -10.

Note that the robot receives rewards not based on its current state (as in the book), but based on state-action-state triples (as in project 4). For this problem, you may assume a discount rate of 0.9.

**a)** Assuming $0 \leq R_{wait} \leq R_{clean}$, which of the following can be optimal behavior (circle all that apply).

(a) Always *clean*, regardless of battery.

(b) Always *recharge*, regarless of battery.

(c) *Clean* with high battery, *recharge* with low battery.

(d) *Recharge* with high battery, *clean* with low battery.

**b)** Write a Bellman equation relating the optimal utility of the state *low* in terms of other optimal utilities. You should use specific variables and probabilities from the problem statement above when possible.

**c)** Let $R_{clean} = 3$ and $R_{wait} = 1$. Assume you have an initial estimate of zero for each state's utility. What are the estimates after one round of value iteration?

**d)** Using the rewards above and a learning rate of 0.2, consider using Q-learning to estimate q-values for this problem. Assume all of your q-value estimates are zero initially. Imagine you observe the following sequence of states, actions, and rewards: high, clean, +3, high, clean, +3, low, clean +3, low, clean -10. Show the q-values for each state after each reward.

*End of Exam*