

NAME: _____ SID#: _____ Login: _____ Sec: _____ 1

CS 188
Fall 2007

Introduction to
Artificial Intelligence

Final Exam

You have 180 minutes. The exam is closed book, closed notes except a two-page crib sheet, basic calculators only. 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.

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.

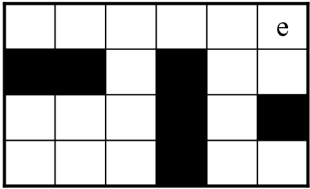
For staff use only

Q. 1	Q. 2	Q. 3	Q. 4	Q. 5	Q. 6	Q. 7	Total
/17	/11	/19	/8	/12	/18	/15	/100

THIS PAGE INTENTIONALLY LEFT BLANK

1. (17 points.) Search and Utilities: Conformant Problems

Consider an agent in a maze-like grid, as shown to the right. Initially, the agent might be in any location x (including the exit), *but it does not know where it is*. The agent can move in any direction (N, S, E, W). Moving into a wall is legal, but does not change the agent's position. For now, assume that all actions are deterministic. The agent is trying to reach a designated exit location e where it can be rescued. However, while the agent knows the layout of the maze, it has no sensors and cannot tell where it is, or even what walls are nearby.



The agent must devise a plan which, on completion, guarantees that the agent will be in the exit location, *regardless of the (unknown) starting location*. For example, here, the agent might execute $[W, N, N, E, E, N, N, E, E, E]$, after which it will be at e regardless of start position. You may find it useful to refer to $pre(x, a)$, the either empty or singleton set of squares which lead to x on a successful action a , and/or $post(x, a)$, the square resulting from x on a successful action a .

(a) (4 points) Formally state this problem as a single agent state-space search problem. You should formulate your problem so that your state space is finite (e.g. do not use an encoding where each partial plan is a state).

States:

Size of State Space:

Start state:

Successor function:

Goal test:

(b) (4 points) Give a non-trivial admissible heuristic for this problem.

Imagine the agent's movement actions may fail, causing it to stay in place with probability f . In this case, the agent can never be sure where it is, no matter what actions it takes. However, after any sequence of actions $a = a_1 \dots a_k$, we can calculate a belief state $P(x|\mathbf{a})$ over the locations x in the grid.

(c) (3 points) Give the expression from an incremental algorithm for calculating $P(x|a_1 \dots a_k)$ in terms of $P(x|a_1 \dots a_{k-1})$. Be precise (e.g., refer to x , f , and so on).

$$P(x|a_1 \dots a_k) =$$

Imagine the agent has a new action $a = Z$ which signals for pick-up at the exit. The agent can only use this action once, at which point the game ends. The utility for using Z if the agent is actually at the exit location is +100, but -1000 elsewhere.

(d) (3 points) If the agent has already executed movement actions $a_1 \dots a_k$, give an expression for the utility of then executing Z in terms of the quantity computed in (c).

$$U(A_{k+1} = Z|a_1 \dots a_k) =$$

Imagine that the agent receives a reward of -1 for each movement action taken, and wishes to find a plan which maximizes its expected utility. Assume there is no discounting. Note that despite the underlying uncertainty, this problem can be viewed as a deterministic state space search over the space of plans. Unlike your answer in (a), this formulation does not guarantee a finite search space.

(e) (3 points) Complete the statement of this version of the problem as a single agent state-space search problem. Remember that state space search *minimizes* cost and costs should be non-negative!

States: partial plans, which are strings of the form $\{N, S, E, W\}^*$ possibly followed by Z

Size of State Space: infinite

Start state: the empty plan

Successor function: append N, S, E, W, or Z if current plan does not end in Z, no successors otherwise

Goal test:

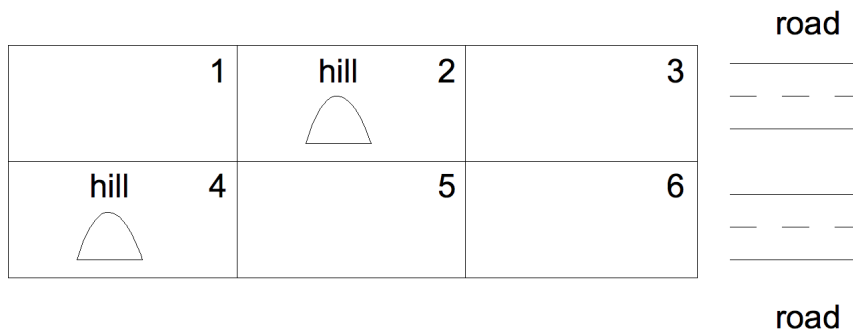
Step cost:

2. (11 points.) CSPs: Layout

You are asked to determine the layout of a new, small college. The campus will have three structures: an administration building (A), a bus stop (B), a classroom (C), and a dormitory (D). Each building must be placed somewhere on the grid below. The following constraints must be satisfied:

- (i) The bus stop (B) must be adjacent to the road.
- (ii) The administration building (A) and the classroom (C) must both be adjacent to the bus stop (B).
- (iii) The classroom (C) must be adjacent to the dormitory (D).
- (iv) The administration building (A) must *not* be adjacent to the dormitory (D).
- (v) The administration building (A) must not be on a hill.
- (vi) The dormitory (D) must be on a hill or near the road.
- (vii) All buildings must be in different grid squares.

Here, “adjacent” means that the buildings must share a grid edge, not just a corner.



(a) (3 points) Express the non-unary above constraints as implicit *binary* constraints over the variables A,B,C,D. Precise but evocative notation such as *different(X, Y)* is acceptable.

(b) (3 points) Cross out eliminated values to show the domains of all variables after unary constraints and arc consistency have been applied (but no variables have been assigned).

$$\begin{array}{l}
 A \quad [\quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad] \\
 B \quad [\quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad] \\
 C \quad [\quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad] \\
 D \quad [\quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad]
 \end{array}$$

(c) (3 points) Cross out eliminated values to show the domains of the variables after $B = 3$ has been assigned and arc consistency has been rerun.

$$\begin{array}{l}
 A \quad [\quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad] \\
 B \quad [\quad \quad \quad 3 \quad \quad \quad \quad] \\
 C \quad [\quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad] \\
 D \quad [\quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad]
 \end{array}$$

(d) (2 points) Give a solution for this CSP or state that none exist.

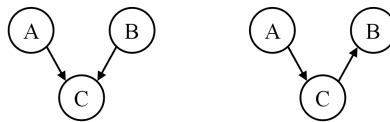
3. (19 points.) Bayes' Nets

Consider the following pairs of Bayes' nets. If the two networks have identical conditional independences, write *same*, along with writing one of their shared independence (or *none* if they assert none). If the two networks have different conditional independences, write *different*, along with writing an independence that one has but not the other. For example, in the following case you would answer as shown:

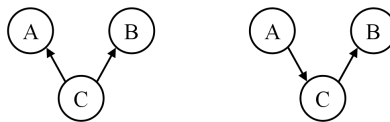


different, right has $A \perp\!\!\!\perp B \mid \{\}$.

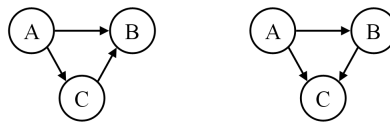
(a) (1 pt)



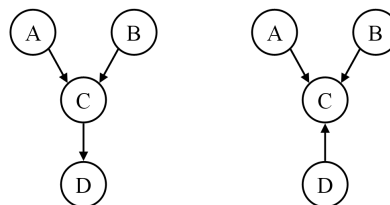
(b) (1 pt)



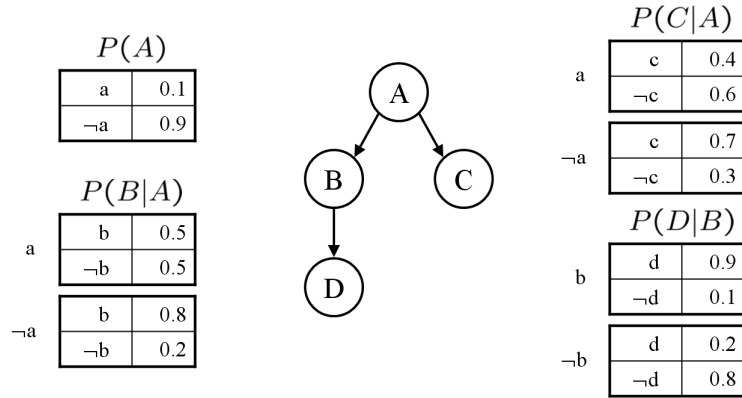
(c) (1 pt)



(d) (1 pt)



The next parts involve computing various quantities in the network below. These questions are designed so that they can be answered with a minimum of computation. If you find yourself doing copious amount of computation for each part, step back and consider whether there is simpler way to deduce the answer.



(e) (2 pts) $P(a, \neg b, c, \neg d)$

(f) (2 pts) $P(b)$

(g) (2 pts) $P(a|b)$

(h) (2 pts) $P(d|a)$

(i) (2 pts) $P(d|a, c)$

Consider computing the following quantities in the above network using various methods:

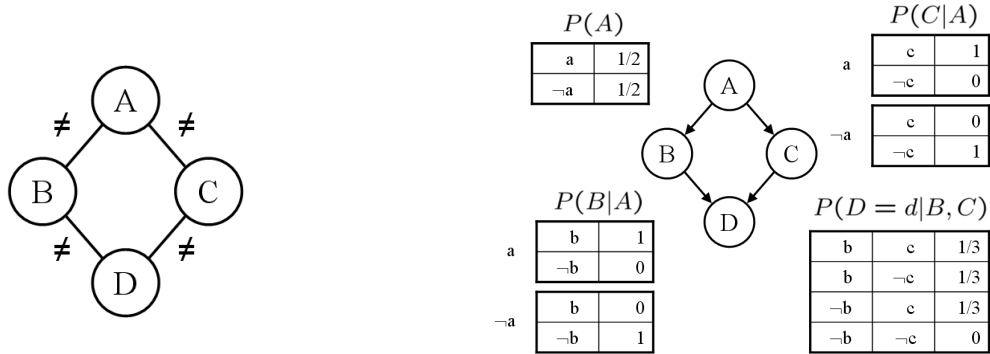
(i) $P(A|b, c, d)$ (ii) $P(C|d)$ (iii) $P(D|a)$ (iv) $P(D)$

(j) (2 pts) Which query is least expensive using inference by enumeration?

(k) (2 pts) Which query is most improved by using likelihood weighting instead of rejection sampling (in terms of number of samples required)?

4. (8 points.) CSPs and Bayes' Nets

The following CSP and Bayes' net will be referred to in the questions below. *They are not equivalent.*



Consider the CSP over 4 binary variables shown on the left. The constraints are that no adjacent variables may be equal (so there are only two legal assignments).

(a) (4 pts) Is there a Bayes' net over the same variables which assigns equal, non-zero probability to assignments which satisfy the CSP but assigns probability zero to assignments which violate the CSP? If so, specify both its graph structure and its CPTs. If not, state why no such Bayes' net can exist.

Consider the Bayes' net shown on the right.

(b) (4 pts) Is there a CSP over the same variables which is satisfied by all and only assignment with non-zero probability in the network? If so, specify its constraint graph structure and its explicit constraints. If not, state why no such CSP can exist.

5. (12 points.) HMMs: Forward and Backward Algorithms

Recall that HMMs model hidden variables $X_{1:T} = X_1, \dots, X_T$ and evidence variables $E_{1:T} = E_1 \dots E_T$. The forward algorithm incrementally computes $P(X_t, e_{1:t})$ for increasing t for the purpose of calculating $P(X_t|e_{1:t})$, the posterior belief over X_t given current evidence e_t and past evidence $e_{1:t-1}$ in an HMM. A more general query is to condition on *all* evidence, past, present, and future: $P(X_t|e_{1:N})$. In this problem, you will work out a method of doing so.

(a) (4 points) Use the laws of probability and the conditional independence properties of an HMM to give an expression for $P(e_{t+1:N}|x_t)$ in terms of $P(e_{t+2:N}|x_{t+1})$ and the basic HMM quantities ($P(X|X')$ and $P(E|X)$). You do not need to worry about the base case.

$$P(e_{t+1:N}|x_t) =$$

This computation is called the backward algorithm.

(b) (3 points) Give an expression for the posterior distribution at a single time step, $P(x_t|e_{1:N})$, in terms of basic HMM quantities and / or quantities computed by the forward and backward algorithms. *Hint: use the chain rule along with the conditional independence properties of HMMs.*

$$P(x_t|e_{1:N}) =$$

(c) (2 points) Give an expression for the posterior distribution over two time steps, $P(x_t, x_{t-1}|e_{1:N})$, in terms of basic HMM quantities and / or quantities computed by the forward and backward algorithms.

$$P(x_t, x_{t-1}|e_{1:N}) =$$

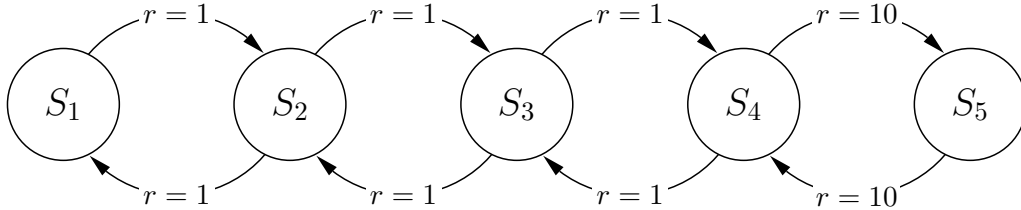
In a second-order HMM, the transition function depends on the past two states: $P(x_t|x_{1:t-1}) = P(x_t|x_{t-1}, x_{t-2})$. Emissions still depend only on the current state.

(d) (3 points) Give the second-order generalization of the forward recurrence. Again, you may disregard the base case. *Hint: you should think about both the left and right hand sides.*

$$P(x_t, x_{t-1}, e_{1:t}) =$$

6. (18 points.) MDPs and Reinforcement Learning

Consider the following Markov Decision Process:



We have states $S_1, S_2, S_3, S_4,$ and S_5 . We have actions *Left* and *Right*, and each action deterministically leads to a successor state (with probability 1). In S_1 , the only available action is to go to S_2 (*Right*), and similarly in S_5 we can only go to S_4 (*Left*). The reward for any action is 1 except for taking *Right* from S_4 and *Left* from S_5 , which have reward 10. Assume a discount factor $\gamma = 0.5$. Recall that $\sum_{i=1}^{\infty} 0.5^i = 1$

(a) (1 pt) What is the optimal policy for this MDP?

$$\{S_1 : \quad , S_2 : \quad , S_3 : \quad , S_4 : \quad , S_5 : \quad \}$$

Value Iteration: When performing value iteration, what is the value of state S_3 after

(b) (1 pt) 1 iteration:

(c) (1 pt) 2 iterations:

(d) (2 pt) ∞ iterations:

Policy Iteration: Suppose you run policy iteration. During each iteration, you compute the exact values in the current policy, then update the policy using one-step lookahead given those values. Suppose, too, that when choosing actions when updating the policy, ties are broken by choosing *Left*.

Suppose we start with the following policy:

$$\{S_1 : \textit{Right}, S_2 : \textit{Right}, S_3 : \textit{Left}, S_4 : \textit{Right}, S_5 : \textit{Left}\}$$

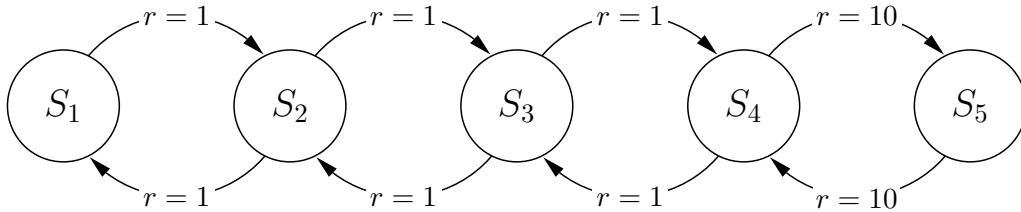
What will be the policy after... (Note: Fill in with R and L for Right and Left)

(e) (2 pt) 1 iteration: $\{S_1 : \quad , S_2 : \quad , S_3 : \quad , S_4 : \quad , S_5 : \quad \}$

(f) (2 pt) 2 iterations: $\{S_1 : \quad , S_2 : \quad , S_3 : \quad , S_4 : \quad , S_5 : \quad \}$

(g) (2 pt) 3 iterations: $\{S_1 : \quad , S_2 : \quad , S_3 : \quad , S_4 : \quad , S_5 : \quad \}$

MDP Repeated here for convenience:



Q-learning: Consider executing Q-learning on this MDP. Assume the learning rate $\alpha = 0.5$, and that Q-learning uses a greedy exploration policy, meaning that it always chooses the action with maximum Q-value. Suppose the algorithm breaks ties by choosing *Left*.

(h) (4 pts) What are the first 10 *(state, action)* pairs visited if our agent learns using Q-learning and starts in S_3 (e.g., $(S_3, Left)$, $(S_2, Right)$, $(S_3, Right)$, ...)?

(i) (3 pts) What is the Q-value of $(S_3, Left)$ after these first 10 actions?

7. (15 points.) Classification and VPI: Catch the Cheater

You work for a casino where the most popular game involves flipping a coin and seeing if it lands heads up. You are in charge of catching cheaters, who use two sided coins, which always come up heads, instead of standard fair coins. Assume that the prior probability of a gambler being a cheater is $1/16$ and that a cheater always uses an unfair coin.

(a) (3 pts) Draw a naive Bayes model over the variables C (cheater) and F_1 to F_N (N consecutive coin flips).

(b) (3 pts) In this model, what is the probability that a gambler is a cheater if you observe 4 flips, all heads?

As a gambler leaves the casino, you can either *accuse* them of having cheated or let them *pass*. Imagine that catching a cheater is worth +5 points, falsely accusing a non-cheater is worth -10, passing on a non-cheater is worth 0, and passing on a cheater is worth -1. For the next problems, you may leave your answers in fractional form.

(c) (2 pts) What is the optimal action if the gambler's record for the night was 4 heads, and what is the expected utility of that action?

(d) (3 pts) What is the probability that a fifth flip would have been heads again?

(e) (4 pts) What is the value of information of the outcome of a fifth flip?