UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN
CS440/ECE448 Artificial Intelligence
Practice Exam 5
Spring 2026

Exam is May 12, 2026, 8:00

Your Name: ______________________________

Your NetID: ______________________________

Instructions

- SHOW YOUR WORK! Correct numerical answers will not receive full credit unless you show sufficient derivation to convince us that you understand the answer.
- Have your ID ready; you will need to show it when you turn in your exam.
- Make sure that your answer includes only the variables that it should include, but DO NOT simplify explicit numerical expressions, including expressions involving standard functions (exp, log, sin, cos) and expressions involving the sum, product, maximum, or argmax of a complete list of alternatives. For example, the answer $x = \max\left(\sin(0.3), \frac{1}{1+\exp(-0.1)}\right)$ is MUCH preferred (much easier for us to grade) than the answer $x = 0.524979$.
- This will be a CLOSED BOOK, CLOSED NOTES exam. You are permitted to bring and use TWO 8.5x11 pages of notes, front and back, handwritten.
- No electronic devices (phones, tablets, calculators, computers etc.) are allowed.

Possibly Useful Formulas

$$P(X=x|Y=y)P(Y=y)=P(Y=y|X=x)P(X=x)$$
$$P(X=x)=\sum_y P(X=x,Y=y)$$
$$E\left[f(X,Y)\right]=\sum_{x,y} f(x,y)P(X=x,Y=y)$$

Precision,Recall=Sensitivity,Specificity $= \frac{TP}{TP+FP}, \frac{TP}{TP+FN}, \frac{TN}{TN+FP}$

MPE=MAP: $f(x)=\arg\max\left(\log P(Y=y)+\log P(X=x|Y=y)\right)$

Naive Bayes: $P(X=x|Y=y)\approx\prod_{i=1}^{n} P(W=w_i|Y=y)$

Laplace Smoothing w/OOV: $P(W=w_i)=\frac{k+\text{Count}(W=w_i)}{k+\sum_v\left(k+\text{Count}(W=v)\right)}$

Fairness: $P(Y|A)=\frac{P(Y|f(X),A)P(f(X)|A)}{P(f(X)|Y,A)}$

Score: $v_{t+1}(j)=\max_i v_t(i)+\log a_{ij}+\log b_j(x_{t+1})$

Backpointer: $\psi_{t+1}(j)=\arg\max_i v_t(i)+\log a_{ij}+\log b_j(x_{t+1})$

Linear Regression: $\mathscr{L}=\frac{1}{n}\sum_{i=1}^{n}\left(\mathbf{w}^T\mathbf{x}_i-y_i\right)^2$

$$\sigma(z)=\frac{1}{1+e^{-z}},\quad \sigma'(z)=\sigma(z)\left(1-\sigma(z)\right)$$
$$f=g\left(\mathbf{w}^T\mathbf{x}\right),\quad \frac{\partial f}{\partial \mathbf{w}}=g'\left(\mathbf{w}^T\mathbf{x}\right)\mathbf{x}$$

$$\text{softmax}(\mathbf{z})=\frac{\exp(\mathbf{z})}{\sum_i \exp(z_i)},\quad \frac{\partial\, \text{softmax}_k(\mathbf{z})}{\partial \mathbf{z}}=\underset{k}{\text{softmax}}(\mathbf{z})\left(\mathbb{1}_k-\text{softmax}(\mathbf{z})\right)$$
$$f[k]=\sum_i w[i]x[k-i],\quad \frac{\partial f[k]}{\partial w[i]}=x[k-i]$$

$$\mathbf{C}=\text{softmax}\left(\frac{\mathbf{QK}^T}{\sqrt{d}}\right)\mathbf{V}$$
$$S:\{\mathscr{V}_P,\mathscr{V}_Q\}\rightarrow\{\mathscr{V}_Q,\mathscr{C}\}\text{ s.t. } S(P)=S(Q)=U$$
$$\frac{x'}{x}=\frac{y'}{y}=\frac{-f}{z}$$

Layer-Wise Relevance Propagation: $R(x_d) = \frac{\frac{\partial \mathscr{L}}{\partial x_d} x_d}{\sum_{d'} \frac{\partial \mathscr{L}}{\partial x_{d'}} x_{d'}} R(\mathscr{L})$

Configuration Space Obstacles: $\mathscr{C}_{\text{obs}} = \{\mathbf{c} : \exists \mathbf{b} : \phi(\mathbf{b}, \mathbf{c}) \in \mathscr{W}_{\text{obs}}\}$

Admissible: $\hat{h}(n) \leq h(n)$

Consistent: $\hat{h}(n) - \hat{h}(m) \leq d(n,m)$

Bellman: $u(s) = r(s) + \gamma \max_a \sum_{s'} P(s'|s,a) u(s')$

Policy Eval: $u_i(s) = r(s) + \gamma \sum_{s'} P(s'|s,\pi_i(s)) u_i(s')$

Policy Update: $\pi_{i+1}(s) = \arg\max_a \sum_{s'} P(s'|s,a) u_i(s')$

Value Iteration: $u_{i+1}(s) = r(s) + \gamma \max_a \sum_{s'} P(s'|s,a) u_i(s')$

Model-based: $P(s_{t+1}|s_t,a_t) = \frac{N(s_t,a_t,s_{t+1}) + k}{\sum_{s'} (N(s_t,a_t,s') + k)}$

Q-Learning: $q(s_t,a_t) \leftarrow (1-\eta) q(s_t,a_t) + \eta q_{\text{local}}(t)$

TD-Learning: $q_{\text{local}}(t) = r_t + \gamma \max_a q_t(s_{t+1},a)$

SARSA: $q_{\text{local}}(t) = r_t + \gamma q_t(s_{t+1},a_{t+1})$

Deep Q: $\mathscr{L}_{\text{critic}} = \frac{1}{2} \left(q_t(s_t,a_t) - q_{local}(t)\right)^2$

Actor-Critic: $\mathscr{L}_{\text{actor}} = -\sum_a \pi_a(s_t) q_t(s_t,a)$

REINFORCE: $\Delta \mathbf{w} = \eta (r - \mu) \sum_t \frac{\partial \log \pi_{a_t}(s_t)}{\partial \mathbf{w}}$

Max: $v = \max\left(v, \max_c \left(\mathbb{E}\left[v(c)\right]\right)\right), \quad \alpha = \max\left(\alpha, \max_c \left(\mathbb{E}\left[v(c)\right]\right)\right), \beta = \text{Unchanged}$

Min: $v = \min\left(v, \min_c \left(\mathbb{E}\left[v(c)\right]\right)\right), \quad \alpha = \text{Unchanged}, \beta = \min\left(\beta, \min_c \left(\mathbb{E}\left[v(c)\right]\right)\right)$

Chance: $\mathbb{E}\left[v\right] = \sum_c P(c) v(c), \quad \alpha = \text{Unchanged}, \beta = \text{Unchanged}$

$$\begin{bmatrix} -1 \\ 1 \end{bmatrix}^T \mathbf{A} \beta = 0, \quad \alpha^T \mathbf{B} \begin{bmatrix} -1 \\ 1 \end{bmatrix} = 0$$

$$a = a + \eta \frac{\partial E[r_A]}{\partial a}, \quad b = b + \eta \frac{\partial E[r_B]}{\partial b}$$

Question 1 *(24 points)*

A spam filter classifies emails as "Spam" or "Not Spam." The prior probability of spam is 0.4.

(a) (12 points) Suppose you pass a piece of e-mail through a pretty good spam filter. The probability the filter flags spam correctly is 0.9. The probability the filter incorrectly flags legitimate email as spam is 0.1. Your email is flagged as spam. Compute the posterior probability that the email is actually spam.

Solution: Define $Y = 1$ to mean spam, $f = 1$ to mean the classifier calls it spam. Then:

$$P(Y=1|f=1) = \frac{P(Y=1)P(f=1|Y=1)}{P(Y=1)P(f=1|Y=1)+P(Y=0)P(f=1|Y=0)}$$
$$= \frac{(0.4)(0.9)}{(0.4)(0.9)+(0.6)(0.1)}$$

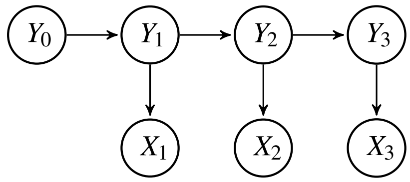
(b) (12 points) Now suppose you don't have a good spam filter, so you must design one. A Naive Bayes classifier calls the email spam if $f(x) > g(x)$, where x is the set of words in the email, and where f and g are functions that you must design. You have a training corpus of 400 spam emails (200 containing the word "win," 50 containing the word "probably") and 600 non-spam emails (60 containing the word "win," 120 containing the word "probably"). In terms of the Laplace smoothing parameter k, what are $f(x)$ and $g(x)$ for an e-mail that contains both "win" and "probably"?

Solution:

$$f(x) = \log P(\text{spam}) + \log P(\text{win}|\text{spam}) + \log(\text{probably}|\text{spam})$$
$$= \log(0.4) + \log\left(\frac{200+k}{400+2k}\right) + \log\left(\frac{50+k}{400+2k}\right)$$
$$g(x) = \log P(\neg\text{spam}) + \log P(\text{win}|\neg\text{spam}) + \log(\text{probably}|\neg\text{spam})$$
$$= \log(0.6) + \log\left(\frac{60+k}{600+2k}\right) + \log\left(\frac{120+k}{600+2k}\right)$$

Question 2 ***(24 points)***

Consider the following hidden Markov model:



Suppose that Y_t and X_t are both binary for all t, with transition and observation probabilities abbreviated as $P(Y_t = j|Y_{t-1} = i) = a_{i,j}$ and $P(X_t = j|Y_t = i) = b_{i,j}$, and suppose it is known that $Y_0 = 0$.

(a) (12 points) In terms of the model parameters, what is $P(X_1 = 1|Y_0 = 0, Y_2 = 1)$?

Solution:

$$P(X_1 = 1|Y_0 = 0, Y_2 = 1) = \frac{\sum_{y_1} P(Y_0 = 0, Y_1 = y_1, X_1 = 1, Y_2 = 1)}{\sum_{x_1, y_1} P(Y_0 = 0, Y_1 = y_1, X_1 = x_1, Y_2 = 1)}$$
$$= \frac{a_{0,0}b_{0,1}a_{0,1} + a_{0,1}b_{1,1}a_{1,1}}{a_{0,0}b_{0,1}a_{0,1} + a_{0,1}b_{1,1}a_{1,1} + a_{0,0}b_{0,0}a_{0,1} + a_{0,1}b_{1,0}a_{1,1}}$$

(b) (12 points) Let $v_t(j)$ and $\psi_t(j)$ be the Viterbi score and backpointer, respectively (the log probability of the best path ending in state j at time t, and the corresponding previous state). Suppose that $v_2(1) = -15$, $\psi_2(1) = 0$, and $X_2 = 1$. In terms of the model parameters, what is $v_1(0)$?

Solution:

$$v_2(1) = -15 = \max_{y_1} v_1(y_1) + \log a_{y_1,1} + \log b_{1,1}$$
$$= v_1(0) + \log a_{0,1} + \log b_{1,1}$$
$$v_1(0) = -15 - \log a_{0,1} - \log b_{1,1}$$

Question 3 *(24 points)*

The softmax function converts real-valued inputs, $-\infty < z_n < \infty$, into non-negative outputs that add up to one. Besides the standard softmax, there are many other functions that accomplish these goals, e.g.,

$$f_n(\mathbf{z}) = \frac{z_n^2}{\sum_m z_m^2} \tag{1}$$

(a) (12 points) For the function $f_n(\mathbf{z})$ in Eq. (1), find $\partial f_n/\partial z_s$ for both $s = n$ and $s \neq n$.

Solution:

$$\frac{\partial f}{\partial z_s} = \begin{cases} -\frac{2z_n^2 z_s}{(\sum_m z_m^2)^2} & s \neq n \\ \frac{2z_n}{\sum_m z_m^2} - \frac{2z_n^3}{(\sum_m z_m^2)^2} & s = n \end{cases}$$

(b) (12 points) Equation (1) could be used to construct a self-attention layer in which a context vector $\mathbf{c}_i$ is constructed from query, key, and value vectors $\mathbf{q}_i$, $\mathbf{k}_t$, and $\mathbf{v}_t$, all of which have dimension d:

$$\mathbf{c}_i = \sum_t \left(\frac{\left(\mathbf{q}_i^T \mathbf{k}_t/\sqrt{d}\right)^2}{\sum_\tau \left(\mathbf{q}_i^T \mathbf{k}_\tau/\sqrt{d}\right)^2} \right) \mathbf{v}_t \tag{2}$$

Express $\partial c_{i,p}/\partial q_{j,r}$ in terms of $\partial f_n/\partial z_s$ for appropriate values of s and n, where $c_{i,p}$ and $q_{j,r}$ are the p^{th} element of $\mathbf{c}_i$ and the r^{th} element of $\mathbf{q}_j$, respectively.

Solution: The part in parentheses is $f(\mathbf{z}^T)$ if $\mathbf{z}^T = \frac{1}{\sqrt{d}}\mathbf{q}_i^T[\mathbf{k}_1, \ldots, \mathbf{k}_T]$. It's s^{th} element is $z_s = \frac{1}{\sqrt{d}}\mathbf{q}_i^T\mathbf{k}_s$. This has no dependence on any $\mathbf{q}_j$ except for when $j = i$; if $j = i$, then $\frac{\partial z_s}{\partial q_{j,r}} = \frac{1}{\sqrt{d}}k_{s,r}$, so

$$\frac{\partial c_{i,p}}{\partial q_{j,r}} = \begin{cases} 0 & i \neq j \\ = \frac{1}{\sqrt{d}} \sum_t \sum_s \frac{\partial f_t}{\partial z_s} k_{s,r} v_{t,p} & i = j \end{cases}$$

Question 4 ***(24 points)***

Suppose you are training a logistic regression network, $\sigma(z) = (1+e^{-z})^{-1}$, with the following unusual loss function with respect to a desired output y:

$$\mathscr{L} = |\sigma(z) - y| \quad (3)$$

(a) (12 points) What is $\partial\mathscr{L}/\partial z$? You may not leave $\sigma'(z)$, the derivative of $\sigma(z)$, in your answer; you must replace it with a simpler form, possibly including $\sigma(z)$.

Solution:

$$\frac{\partial\mathscr{L}}{\partial z} = \text{sign}\left(\sigma(z) - y\right)\sigma(z)(1-\sigma(z))$$

(b) (12 points) Suppose that z is computed by max-pooling a convolutional network with residual connections,

$$z = \max_i \max_j \left(b[i,j] + \sum_m \sum_n w[i-m, j-n]x[m,n]\right) \quad (4)$$

Use layer-wise relevance propagation to express, in terms of $\partial\mathscr{L}/\partial z$, the relevance of each filter weight $w[m,n]$ to the output $\sigma(z)$. Your answer should explicitly include $\partial\mathscr{L}/\partial z$ and may also explicitly include i^* and j^* defined by

$$(i^*, j^*) = \arg\max_i \arg\max_j \sum_m \sum_n w[i-m, j-n]x[m,n] \quad (5)$$

Solution:

$$\begin{aligned} R[m,n] &= \frac{\frac{\partial\mathscr{L}}{\partial z}\frac{\partial z}{\partial w[m,n]} w[m,n]}{\sum_{m',n'} \frac{\partial\mathscr{L}}{\partial z}\frac{\partial z}{\partial w[m',n']} w[m',n']} \\ &= \frac{w[m,n]x[i^*-m, j^*-n]}{\sum_{m',n'} w[m',n']x[i^*-m', j^*-n']} \end{aligned}$$

Question 5 *(24 points)*

Suppose the configuration space and workspace of a robot are related by a forward kinematic function: $\mathbf{w} = \phi(c)$, where c is an integer and $\mathbf{w}$ is a real vector. Suppose that, from any configuration c, the robot can reach any configuration from $c-2$ through $c+2$ in one step. A path from start to goal is a sequence of states $\{c_0, \ldots, c_T\}$ such that the robot spends exactly one second traveling from c_t to c_{t+1} at constant workspace velocity ($\frac{\partial \mathbf{w}}{\partial \tau}$ is constant for $t \le \tau < t+1$), c_0 is the starting state, $\mathbf{g} = \phi(c_T)$ is the goal in the workspace, and T is the (unknown) time required to reach the goal.

(a) (12 points) Suppose you want to find a path that minimizes the mean-squared velocity of the robot,

$$\mathscr{L} = \int_0^T \left\| \frac{\partial \mathbf{w}}{\partial t} \right\|^2 dt, \tag{6}$$

You can find a path that minimizes $\mathscr{L}$ by using Dijkstra's algorithm to search paths through configuration space. How should you define the distance $d(c_t, c_{t+1})$ so that Dijsktra's algorithm finds the desired path?

Solution:

$$\int_0^T \left\| \frac{\partial \mathbf{w}}{\partial t} \right\|^2 dt = \sum_{t=0}^{T-1} \int_t^{t+1} \left\| \frac{\partial \mathbf{w}}{\partial \tau} \right\|^2 d\tau,$$

therefore the distance should be $d(c_t, c_{t+1}) = \int_t^{t+1} \left\| \frac{\partial \mathbf{w}}{\partial \tau} \right\|^2 d\tau$. The robot has constant velocity in the workspace between t and $t+1$, therefore $\frac{\partial \mathbf{w}}{\partial \tau} = \phi(c_{t+1}) - \phi(c_t)$. Transitions from c to $c+3$ are impossible, so we want

$$d(c_t, c_{t+1}) = \begin{cases} \|\phi(c_{t+1}) - \phi(c_t)\|^2 & |c_{t+1} - c_t| \le 2 \\ \infty & |c_{t+1} - c_t| > 2 \end{cases}$$

(b) (12 points) Suppose that $\hat{h}(c) = \|\phi(c) - \mathbf{g}\|^2$, the squared Euclidean distance between $\phi(c)$ and $\mathbf{g}$. Suppose that you put c_0 on the frontier with $g(c_0) = 0$, then iterate the following loop: (1) From the frontier, pop the node with the lowest $g(c) + \hat{h}(c)$, (2) Add c to the explored set, (3) If $\phi(c) = \mathbf{g}$, end the search and report the path. (4) For each node c' that has $d(c, c') < \infty$ and for which c' is not yet in the explored set, add c' to the frontier with $g(c') = g(c) + d(c, c')$. Does this algorithm always find the shortest path? Why or why not?

Solution: This is the A* algorithm with an explored set, so it finds the shortest path only if the heuristic is both admissible and consistent. However, suppose that $\phi(c')$ is halfway between $\phi(c)$ and the goal, then

$$\begin{aligned} d(c, \phi^1(\mathbf{g})) &= d(c, c') + d(c', \phi^{-1}(\mathbf{g})) \\ &= \|\phi(c) - \phi(c')\|^2 + \|\phi(c') - \mathbf{g}\|^2 \\ &= 2 \times \left(\frac{1}{2} \|\phi(c) - \mathbf{g}\| \right)^2 \\ &= \frac{1}{2} \hat{h}(c) \end{aligned}$$

Since $d(c, \phi^{-1}(\mathbf{g})) < \hat{h}(c)$, the heuristic is not admissible. Since all inadmissible heuristics are also inconsistent, it is also inconsistent.

Question 6 ***(24 points)***

Consider an MDP in which the state variable is a real number, $s \in \Re$, and the reward in any state is equal to the value of the state, $r(s) = s$. Suppose there are two possible actions, $a \in \{-1, +1\}$, and suppose the state transition probabilities are

$$P(s'|s,a) = \begin{cases} p & s' = s + 2a \\ q & s' = s - 3a \\ 1 - p - q & s' = s \end{cases}$$

where the parameters p and q are assumed known in part (a), but assumed unknown in part (b).

(a) (12 points) Suppose you start value iteration with initial utility estimates of of $u_1(s) = s$ in iteration $t = 1$. Find $u_2(s)$ as a function of s, p, q, and γ.

Solution:

$$\begin{aligned} u_2(s) &= r(s) + \gamma \max_a P(s'|s,a) u_1(s') \\ &= s + \gamma \max_{a \in \{-1,1\}} (s + a(2p - 3q)) \\ &= \begin{cases} s\left(1 + \gamma(1 + 2p - 3q)\right) & 2p \geq 3q \\ s\left(1 + \gamma(1 + 3q - 2p)\right) & 2p \leq 3q \end{cases} \end{aligned}$$

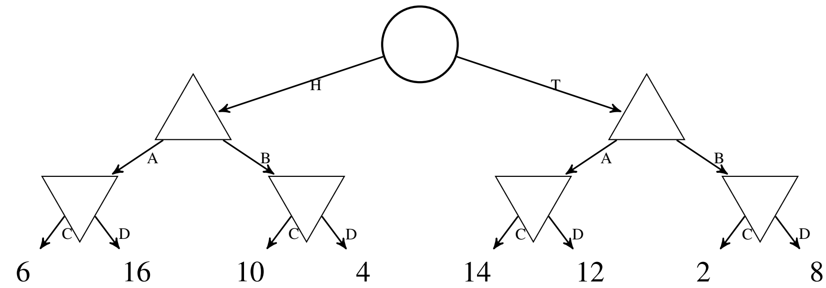
(b) (12 points) Suppose that you are training a policy network for which the probability of action $a = 1$ is $\pi_1(s) = \sigma(\mathbf{w}^T \begin{bmatrix} s \\ 1 \end{bmatrix}$, where $\mathbf{w}^T = [w_1, w_2]$ is a weight vector, $\sigma(\cdot)$ is the logistic sigmoid, and the probability of action $a = -1$ is $\pi_{-1}(s) = 1 - \pi_1(s)$. Suppose the REINFORCE algorithm starts with $\mathbf{w}^T = [0, 0]$, then observes the episode $(s_1, a_1, s_2, a_2) = (0, 1, 1, 1)$ and gains a reward of $r = 5$, which is better than the average reward $\mu = -1$. In terms of the learning rate η, what is the update $\Delta\mathbf{w}$?

Solution:

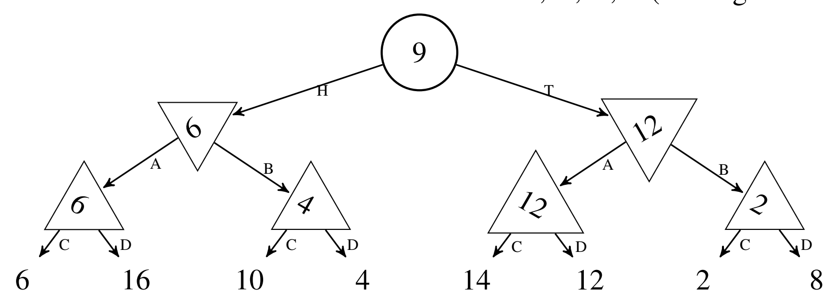
$$\begin{aligned} \Delta\mathbf{w} &= \eta(r - \mu) \sum_t \frac{\partial \log \pi_{a_t}(s_t)}{\partial \mathbf{w}} \\ &= 6\eta \left(\frac{\partial \log \pi_1(0)}{\partial \mathbf{w}} + \frac{\partial \log \pi_1(1)}{\partial \mathbf{w}} \right) \\ &= 6\eta \left((1 - \sigma(0)) \begin{bmatrix} 0 \\ 1 \end{bmatrix} + (1 - \sigma(0)) \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right) \\ &= \begin{bmatrix} 3\eta \\ 6\eta \end{bmatrix} \end{aligned}$$

Question 7 ***(24 points)***

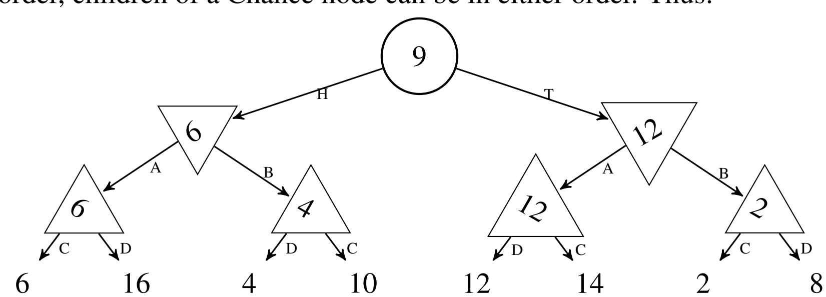
Consider the following game tree. First, Max performs action A or B. Second, Min performs action C or D. Third, a fair coin is flipped, and comes up either H or T. Finally, Max receives the number of points shown in the leaf node, and Min receives 9 minus that number of points.



(a) (12 points) What is the value, to Max, of each of the nodes in the above tree? Write the value of each node inside the corresponding node. What actions should Max and Min perform in each node where they are allowed to act?

> **Solution:** Max chooses A in either node. Min chooses C, D, D, C (reading from left to right).
>
> 
>

(b) (12 points) Suppose you have a heuristic that can tell you which of two game states is more valuable to Max. Assuming the existence of such a heuristic, in what order should the game tree be evaluated by the alpha-beta algorithm so that it only needs to evaluate six of the eight leaf nodes?

> **Solution:** Children of a Max node should be in descending order, children of a Min node in ascending order, children of a Chance node can be in either order. Thus:
>
> 
>

Question 8 ***(32 points)***

Alice and Bob are playing a simultaneous game in which each of them may either cooperate or defect. Their payoffs are shown in the matrix below.

		Bob	
		defects	cooperates
Alice	defects	1 -3	-9 -7
	cooperates	-9 -7	-1 3

(a) (11 points) This game has a mixed equilibrium. What is it?

Solution:

$$\mathbf{A}\beta = \begin{bmatrix} -3 & -7 \\ -7 & 3 \end{bmatrix} \begin{bmatrix} 1-\beta \\ \beta \end{bmatrix} = \begin{bmatrix} -3+4\beta \\ -7+10\beta \end{bmatrix}$$

So the equilibrium probability that Bob cooperates is $\beta = \frac{2}{7}$.

$$\alpha\mathbf{B} = [1-\alpha, \alpha] \begin{bmatrix} 1 & -9 \\ -9 & -1 \end{bmatrix} = [1-10\alpha, -9+8\alpha]$$

So the equilibrium probability that Alice cooperates is $\alpha = \frac{5}{9}$

(b) (10 points) How many Pareto-optimal outcomes does this game have, and what are they?

Solution: There are two: (defect, defect) and (cooperate,cooperate).

(c) (11 points) It is possible to modify just one of the payoffs in this game (just one number in the chart above) so that Bob has a dominant strategy. What number can be modified, to what value, to give Bob a dominant strategy?

Solution: We can give Bob a dominant strategy by modifying the top row so that he always cooperates, or by modifying the bottom row so that he always defects. Thus any of the following answers would be correct:

$$r_B(d,d) < -9$$
$$r_B(d,c) > 1$$
$$r_B(c,d) > -1$$
$$r_B(c,c) < -9$$

This page is scratch paper

This page is scratch paper, you may tear it off if you wish