CS 188
Spring 2014

Introduction to
Artificial Intelligence

Written HW3

INSTRUCTIONS

- **Due:** Tuesday, February 18th, 2014 11:59 PM
- **Policy:** Can be solved in groups (acknowledge collaborators) but must be written up individually. However, we strongly encourage you to first work alone for about 30 minutes total in order to simulate an exam environment. Late homework will not be accepted.
- **Format:** You must solve the questions on this handout (either through a pdf annotator, or by printing, then scanning; we recommend the latter to match exam setting). Alternatively, you can typeset a pdf on your own that has answers appearing in the same space (check edx/piazza for latex templating files and instructions). **Make sure that your answers (typed or handwritten) are within the dedicated regions for each question/part. If you do not follow this format, we may deduct points.**
- **How to submit:** Go to www.pandagrader.com. Log in and click on the class CS188 Spring 2014. Click on the submission titled Written HW 3 and upload your pdf containing your answers. If this is your first time using pandagrader, you will have to set your password before logging in the first time. To do so, click on "Forgot your password" on the login page, and enter your email address on file with the registrar's office (usually your @berkeley.edu email address). You will then receive an email with a link to reset your password.

Last Name	Chen
First Name	Jianzhong
SID	23478230
Email	chenjianzhong@berkeley.edu
Collaborators	None

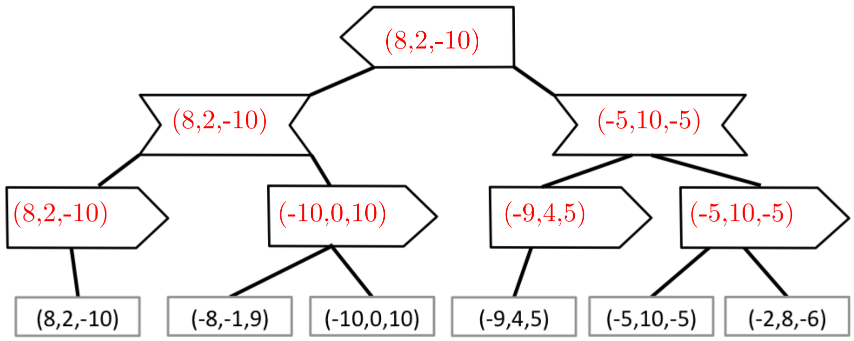
For staff use only

Q. 1	Q. 2	Total
/18	/12	/30

Q1. [18 pts] Games: 3-Player Pruning

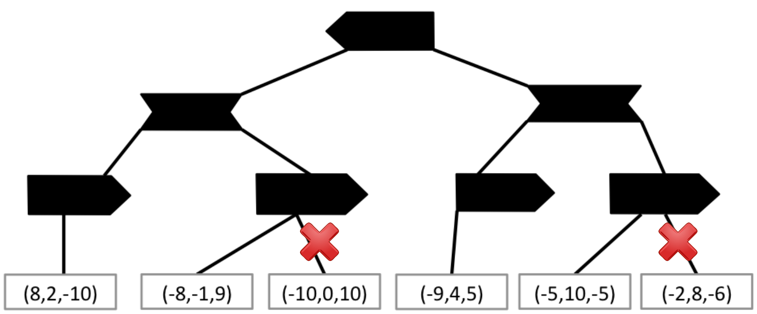
(a) [2 pts] **A 3-Player Game Tree**

Consider the 3-player game shown below. The player going first (at the top of the tree) is the Left player, the player going second is the Middle player, and the player going last is the Right player, optimizing the left, middle and right components respectively of the utility vectors shown. Fill in the values at all nodes. Note that all players *maximize* their own respective utilities.



(b) [3 pts] **Pruning for a 3-Player Zero-Sum Game Tree**

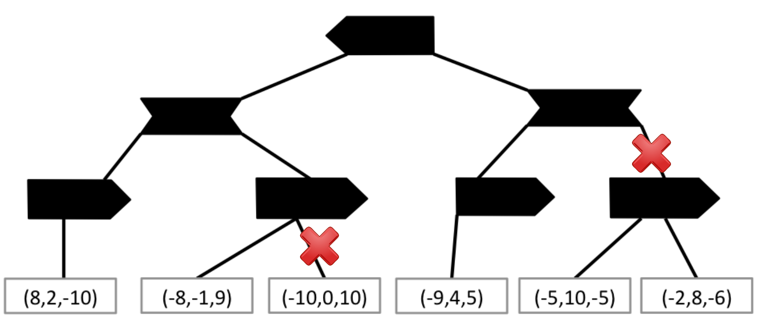
We would like to prune nodes in a fashion similar to α - β pruning. Assume that we have the knowledge that the sum of the utilities of all 3 players is always zero. What pruning is possible under this assumption? Assume the game tree is traversed from left to right. Below, cross off with an $\times$ any branches that can be safely pruned. Justify your answer.



1b explanation: After we exploring (8,2,-10), we know the best option for player2 is currently 2, and the best option for player1 is 8. Then we explore (-8,-1,9), because sum of all three players is always zero, and all utilities are in the interval [-10,10], we can not get a score better than -11(0-9-2=-11) for player1 from the other child of right maximizer, so we prone the other child. Similarly, when we explore the node (-5,10,-5), we know we can not get a score better than 1(0-4-(-5)=1) for player1 from the other child.

(c) [3 pts] **Pruning for a 3-Player Zero-Sum, Bounded-Utility Game Tree.**

If we assume more about a game, additional pruning may become possible. Now, in addition to assuming that the sum of the utilities of all 3 players is still zero, we also assume that all utilities are in the interval $[-10, 10]$. What pruning is possible under these assumptions? Assume the game tree is traversed from left to right. Below, cross off with an $\times$ any branches that can be safely pruned. Justify your answer.



1c explanation: for node ((8,2,-10), the reason is the same as above one. After we explore the node (-9,4,5), we know that we can not get a score better than 6 (0-4-(-10)=6) for player1 from middle maximizer, because the sum of all utilities is 0 and all utilities are in the interval [-10,10]

(d) [6 pts] **Pruning Criteria**

Again consider the zero-sum, bounded-utility, 3-player game described above in (c). Here we assume the utilities are bounded by $[-C, C]$ for some constant C (the previous problem considered the case $C = 10$). Assume there are only 3 actions in the game, and the order is Left, Middle, then Right. Below is code that computes the utilities at each node, but the pruning conditions are left out. Fill in conditions below that prune wherever safe.

(i) [2 pts] **Pruning for Left**

```
Compute-Left-Value(state, C)
  v_L = -infty; alpha = -infty
  for each successor of state:
    (w_L, w_M, w_R) = Compute-Middle-Value(successor, C, alpha)
    if (w_L > v_L)
      v_L = w_L
      v_M = w_M
      v_R = w_R
    end
    // pruning condition:
    if (                  )
      return (v_L, v_M, v_R)
    end
    alpha = max(alpha, v_L)
  end
  return (v_L, v_M, v_R)
```

Fill in the pruning condition below, write $false$ if no pruning is possible:

$(\; v_L == C)$

1d(i) explanation: If v_L is the maximum we can have for utilities, we can prune other children

(ii) [2 pts] **Pruning for Middle**

```
Compute-Middle-Value(state, C, alpha)
  v_M = -infty; beta = -infty
  for each successor of state:
    (w_L, w_M, w_R) = Compute-Right-Value(successor, C, beta)
    if (w_M > v_M)
      v_L = w_L
      v_M = w_M
      v_R = w_R
    end
    // pruning condition:
    if (                  )
      return (v_L, v_M, v_R)
    end
    beta = max(beta, v_M)
  end
  return (v_L, v_M, v_R)
```

Fill in the pruning condition below, write $false$ if no pruning is possible:

$(\; C - v_M < alpha)$

1d(ii) explanation: Because the minimum value we can have for v_R is -C, the maximum value we can have for v_L in middle maximizer is $0 - v_L - (-C) = C - v_L$

(iii) [2 pts] **Pruning for Right**

```
Compute-Right-Value(state, C, beta)
  v_R = -infty;  gamma = -infty
  for each successor of state:
    (w_L, w_M, w_R) = Evaluate-Utility(successor)
    if (w_R > v_R)
      v_L = w_L
      v_M = w_M
      v_R = w_R
    end
    // pruning condition:
    if (                  )
      return (v_L, v_M, v_R)
    end
    gamma = max(gamma, v_R)
  end
  return (v_L, v_M, v_R)
```

Fill in the pruning condition below, write $false$ if no pruning is possible:

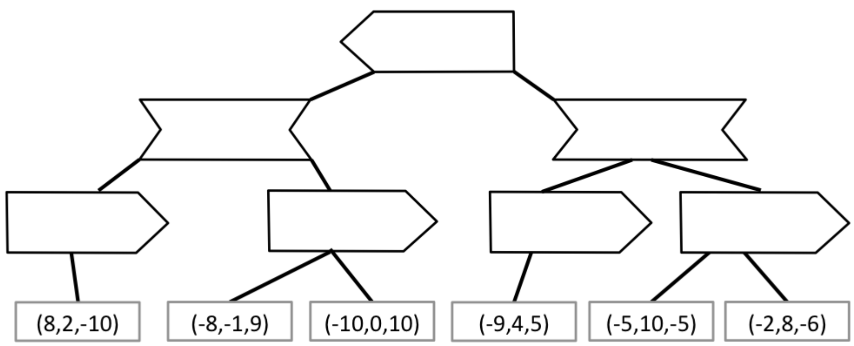
($c - v_R < beta$)

1d(iii) explanation: Because the minimum value we can have for v_L is -C, the maximum value we can have for v_M in the right maximizer is $0 - v_R - (-C) = C - v_R$

(e) [4 pts] **Preparing for the Worst (Case)**

Consider again the game tree from earlier, shown again below. Assume it is not known whether Middle and Right are rational. Left wants to avoid the worst-case scenario according to its own utilities.

Note: you do not need to fill in this game tree; it is provided purely to aid you in answering the question below.



(i) [2 pts] What is Left's optimal action? What is the utility achieved in this worst-case scenario?

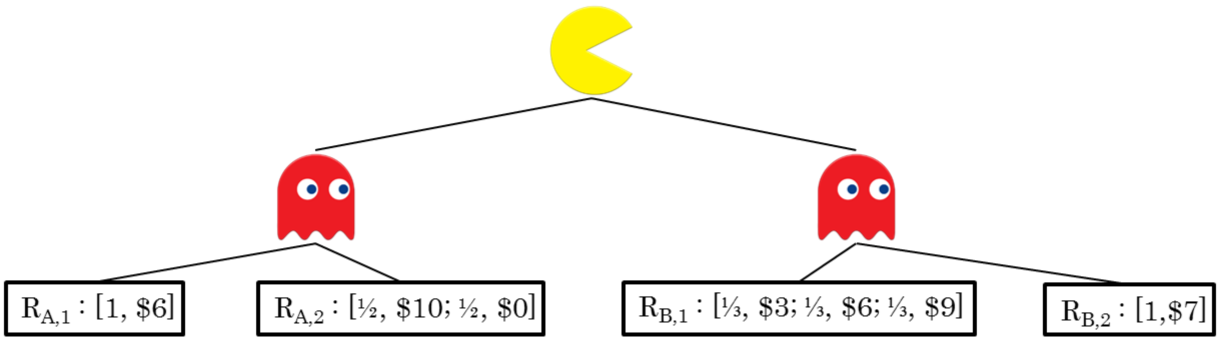
1(e) i answer: Right is the optimal action. In the worst case, the utility achieved is -9

(ii) [2 pts] Which one of the following approaches would allow, for a generic 3-player game, the Left player to compute the strategy that optimizes its worst-case outcome? (you may ignore pruning here)

- ○ Running the strategy described in the pseudo-code from question (d).
- ● Running standard minimax, considering Left's utility only, considering Left to be a maximizer and considering both Middle and Right to be minimizers.
- ○ Running standard expectimax, considering Left's utility only, considering Left to be a maximizer and considering both Middle and Right to be chance nodes.
- ○ None of the above.

Q2. [12 pts] Utilities

Pacman is buying a raffle ticket from the ghost raffle ticket vendor. There are two ticket types: A and B, but there are multiple specific tickets of each type. Pacman picks a ticket type, but the ghost will then choose which specific ticket Pacman will receive. Pacman's utility for a given raffle ticket is equal to the utility of the lottery of outcomes for that raffle ticket. For example, ticket $R_{A,2}$ corresponds to a lottery with equal chances of yielding 10 and 0, and so $U(R_{A,2}) = U([\frac{1}{2}, 10; \frac{1}{2}, 0])$. Pacman, being a rational agent, wants to maximize his expected utility, but the ghost may have other goals! The outcomes are illustrated below.



(a) [5 pts] Imagine that Pacman's utility for money is $U(m) = m$.

(i) [2 pts] What are the utilities to Pacman of each raffle ticket?

$U(R_{A,1}) = 6$ $\qquad$ $U(R_{A,2}) = 5$

$U(R_{B,1}) = 6$ $\qquad$ $U(R_{B,2}) = 7$

(ii) [1 pt] Which raffle ticket will Pacman receive under optimal play if the ghost is trying to minimize Pacman's utility (and Pacman knows the ghost is doing so)? (circle one)

$R_{A,1}$ $\qquad$ $R_{A,2}$ $\qquad$ $\boxed{R_{B,1}}$ $\qquad$ $R_{B,2}$

(iii) [2 pts] What is the equivalent monetary value of raffle ticket $R_{B,1}$?

2a(iii) answer: 6 dollars

(b) [5 pts] Now imagine that Pacman's utility for money is given by $U(m) = m^2$.

(i) [2 pts] What are the utilities to Pacman of each raffle ticket?

$U(R_{A,1}) = 36$ $\qquad$ $U(R_{A,2}) = 50$

$U(R_{B,1}) = 42$ $\qquad$ $U(R_{B,2}) = 49$

(ii) [1 pt] The ghost is still trying to minimize Pacman's utility, but the ghost mistakenly thinks that Pacman's utility is given my $U(m) = m$, and Pacman is aware of this flaw in the ghost's model. Which raffle ticket will Pacman receive? (circle one)

$R_{A,1}$ $\qquad$ $\boxed{R_{A,2}}$ $\qquad$ $R_{B,1}$ $\qquad$ $R_{B,2}$

(iii) [2 pts] What is the equivalent monetary value of raffle ticket $R_{B,1}$?

2b(iii) answer: 6 dollars

(c) [2 pts] Pacman has the raffle with distribution [0.5, \$100; 0.5, \$0]. A ghost insurance dealer offers Pacman an insurance policy where Pacman will get \$100 regardless of what the outcome of the ticket is. If Pacman's utility for money is $U(m) = \sqrt{m}$, what is the maximum amount of money Pacman would pay for this insurance?

2c answer: 5 dollars