Homework 7
Game Theory

CMU 15-780: Graduate AI (Spring 2016)
OUT: April 19th, 2016
DUE: 11:59pm April 28th, 2016

Instructions

Collaboration/Academic Policy

You may discuss assignments with other students as you work through them, but writeups must be done alone. NO DOWNLOADING OR COPYING OF CODE OR OTHER ANSWERS IS ALLOWED. If you use a string of at least 5 words from some source, you must cite the source; however you cannot just copy the solution from some source but instead you should write it up in your own words.

Submission

Please create a tar archive of your answers and submit to Homework 7 on autolab. You should have two files in your archive: a completed `cfr.py` for the programming portion, and a PDF for your answers to the written component. Your completed functions will be autograded by running through several test cases and their return values will be compared to the reference implementation. Please put your Andrew ID somewhere on the first page of your written answers.

You have 8 late days for homeworks over the semester, and can use at most 3 for one homework.

TAs

If you need help, the names beside the questions are the names of the TAs who came up with them, and are more likely to be familiar with the topics.

1 Written: Equilibrium refinement [Christian, Guillermo; 34 points]

Consider the extensive-form game given in Figure 1. It has three players: $P1, P2, P3$. The utility at leaf nodes is denoted as u_1, u_2, u_3, where u_i is the utility for player i of reaching that leaf node.

1.1 Perfect Bayes-Nash Equilibrium, 17 points

Show a perfect Bayes-Nash equilibrium that is not a sequential equilibrium. State the associated beliefs at the non-singleton information set. If there is more than one possible associated belief, state all of them.

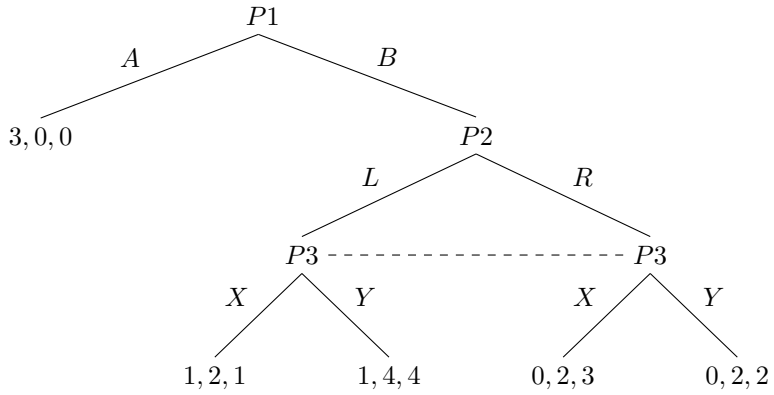

Figure 1: An extensive-form game

1.2 Sequential Equilibrium, 17 points

Show a sequential equilibrium of the same game. State the associated beliefs at the non-singleton information set.

2 Programming: Counterfactual Regret Minimization [Christian, Guillermo; 66 points]

You will be implementing the Counterfactual Regret Minimization (CFR) algorithm. Your job is to fill out the `solve_game` method in `cfr.py`. Feel free to add any helpers methods or classes in `cfr.py` as you see fit.

We are providing the file `game.py` for representing the game state. This file contains the `Game` class, which contains a number of helper methods. These will be used for iterating over the game tree. The class also contains a number of print functions that can help you debug and examine the game properties. If you look at the bottom of the `game.py` file, you will see hard-coded equilibrium strategies for the two example games that we are providing. These may be helpful in debugging as well.

The format that you will be using for strategies is described in `cfr.py`.

The two files `coin.txt` and `kuhn.txt` contain a textual representation of two simple extensive-form games. Kuhn is described here: https://en.wikipedia.org/wiki/Kuhn_poker. The value of the game in Kuhn is -0.0555556 and the value of the game in Coin is 0.375. `game.py` contains functions that can print out the tree representation as well as other information.

We will be testing your algorithm by running it for a large number of iterations on some unseen game files and checking that your algorithm converges to the value of the game.

In addition to the course slides, you may find the original CFR paper helpful, it can be found here: http://www.cs.ualberta.ca/~bowling/papers/07nips-regretpoker.pdf. A gentler introduction to CFR can be found here http://modelai.gettysburg.edu/2013/cfr/cfr.pdf. The gentler introduction has pseudocode for a sampling variant of CFR. Keep in mind that we want you to implement the simpler, original CFR algorithm that does not do sampling.