CS 4710: Artificial Intelligence **Spring 2020**

Homework 2

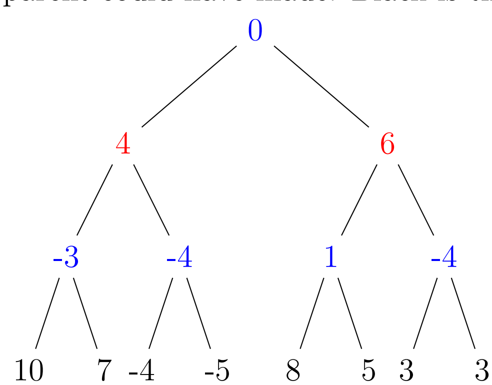
Student Name: David Chen *Computing ID: dzc5ta*

- You will need to submit your solutions in PDF to UVA-Collab.

1. An evaluation function could calculate a score for each possible move based on closest path to ghosts(g1) and (g2), closest path to food (f), and whether or not it would be in a hallway (h). This function could be (g1) + (g2) - (f) - (h). That way a larger distance from the ghosts would yield a better score. A smaller distance to food would also yield a better score. When everything else is the same, the it will prefer not being in a hallway so it can have more options the next turn.

2. A) In minimax, The utility obtained by MAX finds the optimal max assuming that the opponent (MIN) is optimal as well. If the minimax MAX plays against a suboptimal MIN, the MIN will might choose a move that doesn't minimize the score of MAX. Since MAX is optimal, the end score against a sub-optimal MIN is guaranteed to be greater than or equal to its score if it played against an optimal MIN. Again, this is because the sub-optimal MIN is not guaranteed minimize the value of MAX.

   B) Each value is the current score of the game state, blue is trying to maximize the score while red is trying to minimize it. Each child is a possible decision the parent could have made. Black is the end game state.



3 We can use 2 Maps. First, we have P(Xi,di), which contains all consistent values (Xj,dj) in each of Xi's neighbors such that all pairs (di,dj) are consistent. In order to fill this map, we would loop through each variable (Xi) and its values in it domain (di). Within this loop, we loop through all other variables (Xj) and their domains (dj), seeing if they are consistent. If they are consistent, add it to the list of consistent values for P(Xi,di) This will take $O(n^2d^2)$ time because we loop all X values and d values twice. The other Map, C(Xi,di,Xj) will be the number of consistent domain values dj $\in$ Xj that are consistent with di. This can be populated in $O(n^2d^2)$ because for each Xi, di, and Xj, we loop through the values of the domain of Xj in order to count the correct value. We can use these two maps to shorten the runtime of arc

consistency. For checking for consistency, instead of looping through both domains, We can simply loop through C(Xi,di,Xj), to see if there are any 0 values. If there are any, we decrease the corresponding value and put the arc back into the queue.