COMPSCI 370 **Instructor:** Ronald Parr

Homework 2 **Due:** Monday, February 23, 2023

(PDF updated 2/20/23, 4:30 PM)

1 Alpha-beta Warm Up (10 points)

Consider the simple tree from page 9 of the slides, but flip the max and min nodes, i.e., make the root node a min node, and so on. a) Which branches can be pruned? (3 points) b) Explain in English why this is the case. (3 points) c) You can change the value of one node to double the amount of pruning possible. Which node is it, and what is the *smallest* value of this node that will have that effect - assuming that the pruning test uses $\geq$ and $\leq$ as shown in the slides. (4 points)

2 Pruning Chance Nodes (10 points)

Suppose we have a problem (such as backgammon) where a random event occurs before each player moves. In this case, the search tree can be thought of as starting with a max node, followed by a layer of chance nodes, followed by a layer of min nodes, which are then followed by another layer of chance nodes. These chance nodes are followed by max nodes, and the pattern repeats.

If leaf values and evaluation function values are bounded between $L_{\min}$, and $L_{\max}$, what pruning can be done at chance nodes? Write down an if-then statement for pruning similar to what we have in the alpha-beta pseudo-code. To do this precisely, you may assume the following:

- Chance node called with alpha and beta values.
- The number of children of the current chance node is n.
- Your code is looping over these children using a variable i that ranges from $1 \ldots n$.
- The variable v is used to compute a running sum (weighted by probability) of the children seen so far. For example, when $i = 2$, v will contain the weighted sum of the values of the first two children of the chance node.
- You can also assume that you have P_t, which is the sum of the probabilities of the children seen so far.

Your answer should look something like: "If XXX then return YYY" where you need to fill in XXX and YYY. Note that you may not need to use every single piece of information in the above bullets. Does this condition change depending upon the type of parent the node has?

3 Evaluation Functions (10 points)

One of the nice properties of evaluation functions is that they can be useful even if they aren't perfect. To see how this works out, we will consider two trees, $T1$ and $T2$, and assume that $T1$ is the "correct" tree. Suppose $T2$ is created by copying and modifying $T1$ so that for any leaf in $T1$ with value x, the corresponding leaf in $T2$ has value $ax + b$, where $a > 0$ is some constant, and b is an arbitrary constant. a) Provide a simple, inductive proof that for any node at any point in the tree, if the minimax value in $T1$ is y, then the corresponding node in $T2$ has value $ay + b$. (5 points) b) Prove that the optimal action at the root is the same in both $T1$ and $T2$. (5 points)

Once you have proved this, you should have a better understanding of why evaluation functions are useful even if they don't provide the true value that would be calculated if the tree were fully explored to the leaves; the values can be skewed without changing the optimal action at the root. In fact, you have even more flexibility than this. We won't ask you to prove it, but you can probably see how to generalize your proof from affine transformations to any order-preserving transformation. (For evaluation function e_1 for $T1$ and e_2 for $T2$, we would describe e_2 as order preserving if for any pair of leaves a and b, $e_1(a) \geq e_1(b) \leftrightarrow e_2(a) \geq e_2(b)$.)

4 Bayes Rule (10 points)

Some people may be confused about how the following both be true:

- Vaccination reduces the risk of infection.
- Most people who are infected are vaccinated.

Let's look at some numbers to see how this can be. Suppose that the probability of getting a disease if you are vaccinated is reduced by a factor of 4, i.e., $P(d|v) = \frac{1}{4} \times P(d|\overline{v})$, $P(d|\overline{v})$ is 0.4, and $P(d) = 0.15$. Use these to compute $P(v|d)$.