

# CSE471 - Homework 3

Kumal Patel

February 17, 2021

Exercise 1.1 (a) Applying alpha-beta pruning figure 1

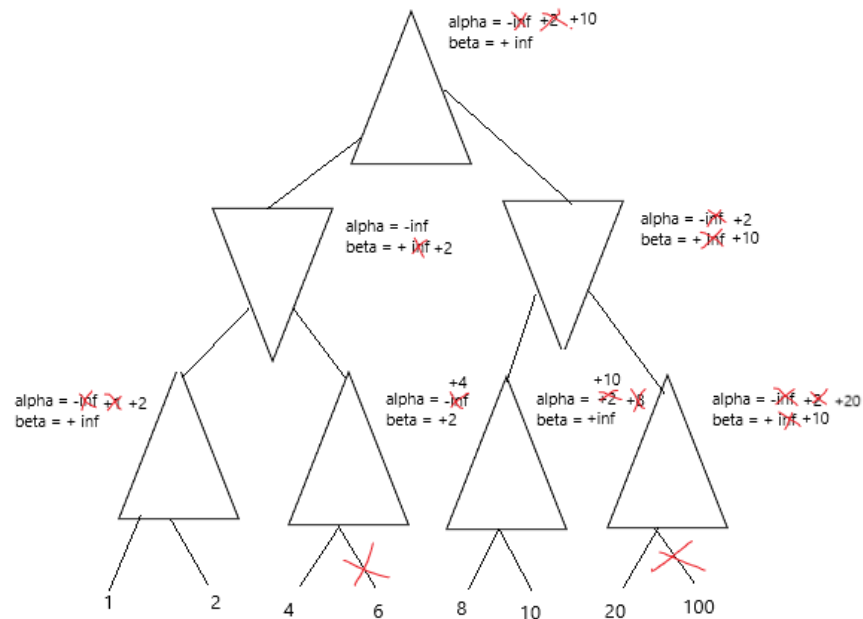


Figure 1: Shows alpha-beta values at each node

(b) Applying alpha-beta pruning figure 2

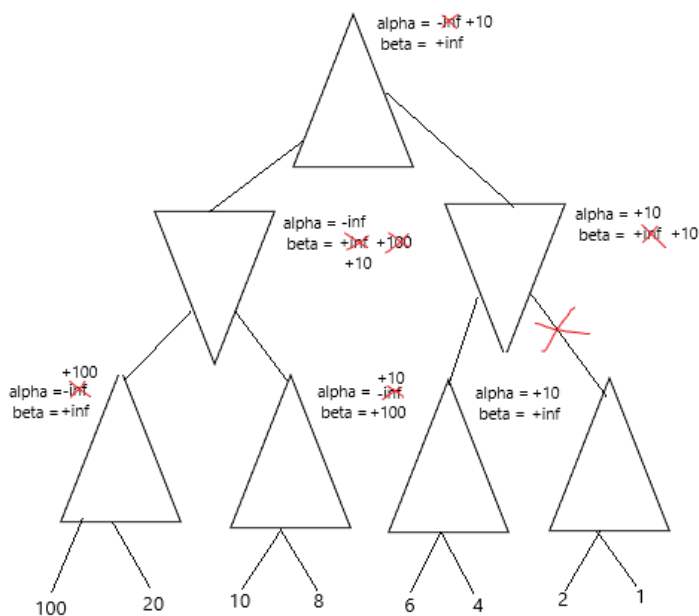


Figure 2: Shows alpha-beta values at each node

(c) Applying alpha-beta pruning figure 3

The ordering of the leaf nodes matter. Pruning occurs when  $\alpha \geq \beta$ . So at f and g (at node 100 and 8),  $\alpha = -\infty$  and  $\beta = 10$ . Since the first leaf node is 100, and  $100 \geq 10$ , we prune. But if 8 was the first leaf it would not prune because  $8 \not\geq 10$ .

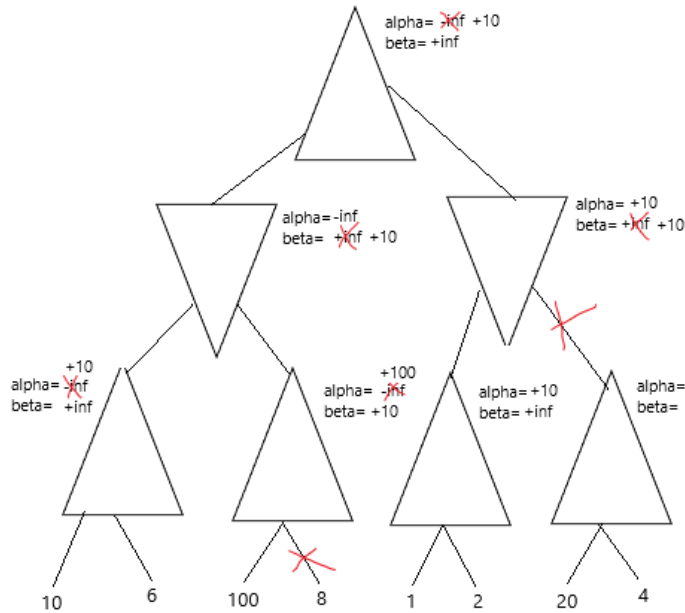


Figure 3: Shows alpha-beta values at each node

- Exercise 1.2 (a) To ensure X1 and its leaf nodes will not be pruned  $A, B, C$  can be 8, 8, 500. Since the max node above X1 has an alpha value of 7. To ensure X1 doesn't prune beta would need to be greater than 7. The value of beta is determined from A or B. And lastly, at node X1, C would need to be greater than A and B to ensure beta is greater than alpha. Choosing a large number like 500 for example would work.
- (b) At node n1 to prune  $A, B, C$  can be 8, 8, 5. The values of A and B can remain the same from part a. C on the other has to be less than or equal to 7 to prune. Since the value of alpha at node X1 is 7.
- (c) At node n2 to prune  $A, B, C$  can be 5, 5, 500. The value of C can

remain the same from part a. A or B on the other hand has to be greater than or equal to 0 to prune. Since the leaf node is 0, beta becomes 0, and the value of alpha would need to be greater or equal to 0 to prune.

- Exercise 1.3
- (a) When the leaf values are finite and unbounded pruning is not possible in a max tree. Because the other branch may contain a higher leaf value than the leaf value that was visited before.
  - (b) Pruning is not possible under the same conditions because for the same reasons as A. The expectimax node must be computed entirely as well for the max node above.
  - (c) If the leaf values are all nonnegative, pruning is not possible in a max tree. Because the root value of a max tree is the best leaf value. And any unvisited leaf value may be the best leaf value, so therefore, all the leaves must be seen.
  - (d) Pruning is not possible in a expectimax tree when all the leaf values are nonnegative. Because the nonnegative values show the lower bound of the chance node. Since pruning is not allowed on the lower bound, pruning is not possible.
  - (e) If the leaf values are in the range of  $[0,1]$  it can be pruned in a max tree.

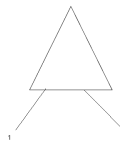


Figure 4: Example showing that the second leaf can be pruned

- (f) If the leaf values are in the range of  $[0,1]$  it can be pruned in an expectimax tree.

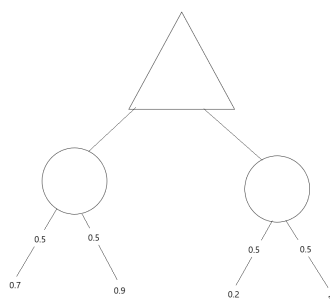


Figure 5: Example showing that the right most leaf can be pruned

- (g) The highest probability first evaluation order is most likely to yield when pruning. Because this makes the strongest bound on the value of the node.

Exercise 1.4 (a)

$$\sum_{i=0}^M U_i(s) = C$$

(b)

$$\sum_{i=0}^M -C * U_i(s)$$

$$\sum_{i=0}^M C * U_i(s)$$

(c)

$$\sum_{i=0}^{M/2} U_i(s) = \sum_{M/2+1}^M U_i(s)$$

- Exercise 1.5
- (a) False, in a zero-sum game with two perfectly rational players, the players would have the perfect knowledge of the game and environment. And in a fully observable environment the player always knows what state it is in. So therefore, knowing all of this helps the first player to be able to predict the second players strategy after making its move.
  - (b) True, in a partially observable environment the player does not know which state it is in. And without further information about the environment it would be difficult to predict the move from the second player. So knowing the second players move would not help in this case.
  - (c) False, even though this agent is perfectly rational it may fail in some unlucky cases. When two agents that are perfectly rational and are playing against each other one must lose, so therefore it is false that a perfectly rational pacman never loses.