CS151 - Written Problem 2
Solutions

1. Designing heuristics

   A knight moves on a chessboard two squares up, down, left, or right followed by one square perpendicular (i.e., the move is L-shaped.) Suppose the knight is on an unbounded board at square $(0, 0)$ and we wish to move it to square (x, y) in the smallest number of moves. For example, to move from $(0, 0)$ to $(1, 1)$ requires two moves.

   (a) Explain how to decide whether the required number of moves is even or odd without constructing a solution.

   Consider the row and column offset between the knight's current position and the goal position. If the parity of both offsets is the same (both even or both odd) the number of moves is even. If the parity is different, the number of moves is odd. The reason for this rule is that the parity between the row and column offset of a single move is different, thus, to make the parity the same, the night would need an even number of moves.

   (b) Design an *admissible* heuristic function for estimating the minimum number of moves required; it should be as accurate as you can make it. Prove rigorously that your heuristic is admissible.

   One admissible heuristic is:

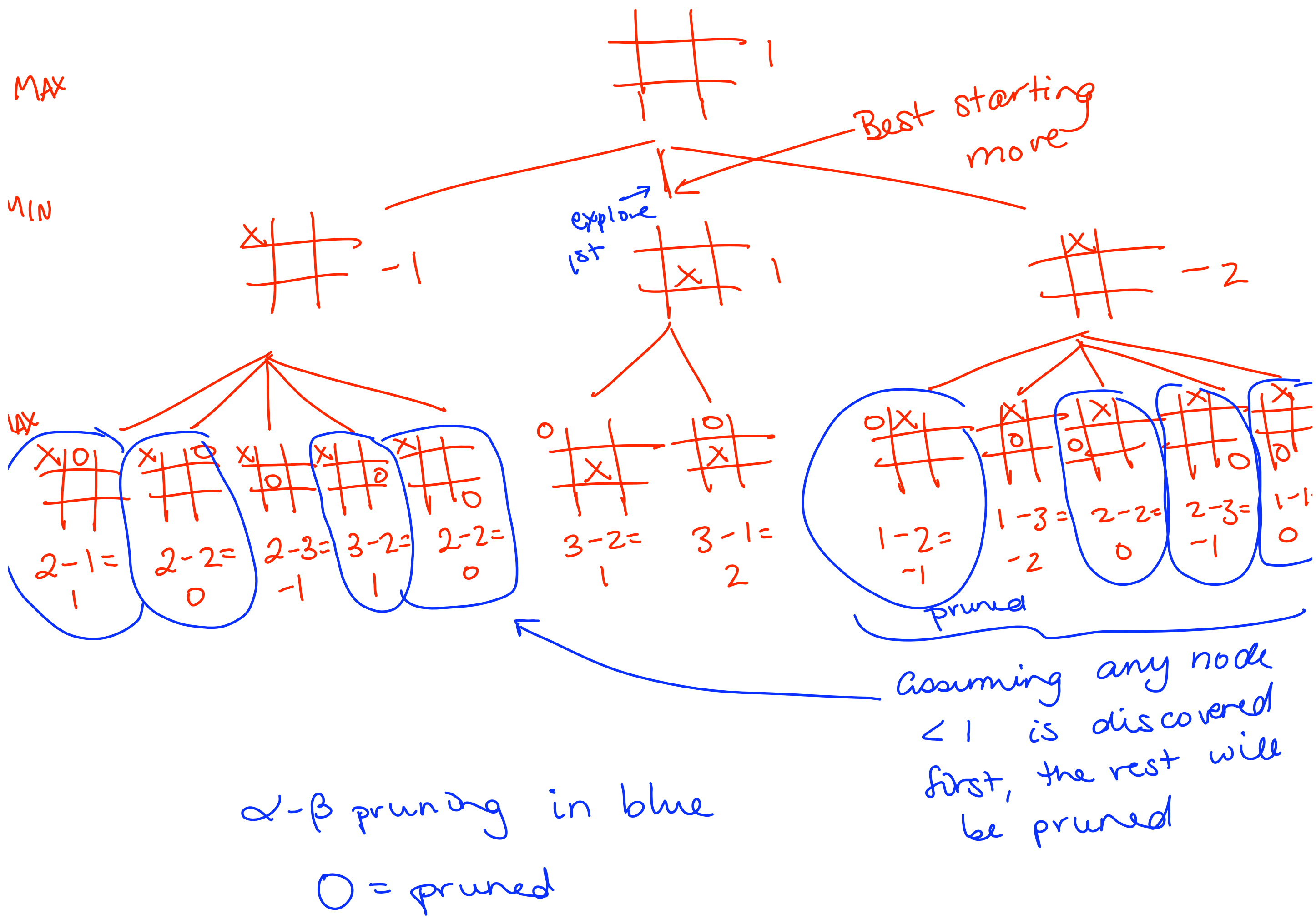
$$h(state) = \max(|x|/2, |y|/2)$$

   where x and y are the offsets described above. This heuristic is admissible because it calculates the number of moves needed when the knight moves maximally in one direction to get to the correct row or correct column, whichever is further away. In other words, there is no way the knight can cover this distance with fewer moves. It does not consider the other direction, and may need more moves to reach the goal.

2. Exercise 5.9

   solution below

```
|
|
|
|
|
|
|
|
|
|
|
|
|
V
```


MAX
MIN
MAX
1
-1
1
-2
Best starting move
explore 1st
2-1=
1
2-2=
0
2-3=
-1
3-2=
1
2-2=
0
3-2=
1
3-1=
2
1-2=
-1
1-3=
-2
2-2=
0
2-3=
-1
0
pruned
Assuming any node < 1 is discovered first, the rest will be pruned
α-β pruning in blue
O = pruned

3. Exercise 5.10 (a-c)

   a. $O(N!)$. At the first level, we have N options, then $N-1$ options at the next level, etc.

   b. All terminal nodes will be at depth N, since the only way to finish a game will be a draw, which will occur when all the spaces have been filled on the board.

   c. A generalization for the Tic Tac Toe we saw in class would work reasonably well. For each player, calculate the number of possible winning subsets that are still obtainable. The evaluation function is the difference in these for the different players.