1. **Describe the possible states, initial state, transition function.**
2. **Describe the terminal states of both checkers and tic-tac-toe.**

The terminal state for checkers is when one of the players has no pieces left so either A wins or B wins, or when the opponent has no viable moves (the other player has pinned them in). The game is a draw if neither player captures their opponent’s piece in 50 total moves.

TTT terminal states is when a player gets 3 (or four) in a row, i.e. either A wins or B wins. There are also draw states when the board is full but no one achieves 3 in a row.

TTT3D’s terminal state is where a player gets four in a row in any of the 3D or 2D planes possible. The draw is the same as TTT.

1. **Why is ν(A,s) = #{white checkers} − #{red checkers} a valid heuristic function for checkers (knowing that A plays white and B plays red)?**

It is a valid heuristic if A plays white and B plays red as generally in checkers if you have more pieces than your opponent left then you are more likely to win which is was a heuristic is supposed to calculate, a value that corresponds how good that move will be for your odds to win. It will also always underestimate the utility function since the difference of pieces left is a simplification. For example, one player having all kings left would be an advantage not considered in this heuristic.

1. **When does v best approximate the utility function, and why?**

v best approximates the utility function when the players have the same piece alignment (i.e. they have the same number of kings, and their pieces can make similar moves).

1. **Can you provide an example of a state s where v(A,s)>0 and B wins in the following turn? (Hint: recall the rules for jumping in checkers)**

Its plausible that there are more white checkers than red checkers and for player B to win when the white checkers are all one tile away from each other (in a “chain”) and a red checker is next to the white checker on the end of the “chain”. Hence the red checker can jump over each white checker in the same turn and win.

1. **Will η suffer from the same problem (referred to in the last question) as the evaluation function ν?**

η should be able to anticipate the double jump moves since it creates a tree to look ahead into each possible move before it makes it until it reaches a winning state.