# MTH 457, Homework

Wil Langford

November 24, 2009

## Homework 1, Problem 3

**Problem** Draw out the game tree for tic-tac-toe. Hint: consider symmetry.

**Solution** The attached PDF is a representation of the game tree for tic-tac-toe.

**Citations**

- Conversations with Aaron Spindel that occured between writing the program and this writeup helped me organize my thoughts.
- A conversation with Prof. John Caughman helped me understand in more detail what was happening when I collapsed symmetrically equivalent game states.
- The wikipedia article on Game Trees was helpful.
- Many online references on Perl, dot and Graphviz were used.
- I tweaked the writeup a little after speaking to Prof. Bleiler.

## Explanation of PDF

The PDF accompanying this writeup is not the game tree for tic-tac-toe. The full game tree for tic-tac-toe is very large and would be difficult to represent as a whole in a meaningful way. Therefore, I've taken a cheap shortcut by creating a digraph that *represents* the game tree for tic-tac-toe.

In the digraph, each vertex represents a symmetrically distinct game state of tic-tac-toe. Game states to the left are parent states of their children on the right. Each edge in the graph represents a symmetrically distinct legal move. Red edges with triangular ends represent corner moves. Green edges with half-box ends represent edge moves. Blue edges with circular ends represent center moves.

A single play of tic-tac-toe can be traced from left to right on the digraph, beginning at the empty board on the far left and ending at a game state with no children (which corresponds to a "leaf" vertex in the full game tree.)

Each game state with more than one parent can be viewed as the root vertex of a symmetric equivalence class of subgames. Therefore, the full game tree could be generated by creating a separate copy of each vertex $v$ for each of $v$'s parents.

There are 765 symmetrically distinct game states in tic-tac-toe. There are 138 "leaves." (That is, 138 distinct win, loss or draw states.)

Symmetric equivalence between two game states was decided based on applying all eight symmetries of a square (from group theory) to one of the game boards and comparing the results to the other board to see if they were equal.

In the primary data structure I used, I referred to each game state only according to a symmetrically equivalent "master state" for its class. Effectively, I chose one game state from each equivalence class as the "name" of the entire class and used it throughout to refer to that class. This approach was a deliberate tradeoff: I wanted condensed enough output to be able to view as a whole, but the ability to trivially trace a single play of tic-tac-toe was lost (as discussed above.) It's still possible to trace a given play, but it's not trivially easy.