CMPT 310, Spring 2019, Written Assignment 03

Due date: January 28, 2019

Problem 1. This problem exercises the basic concepts of game playing, using tic-tac-toe (noughts and crosses). We define X_n as the number of rows, columns, or diagonals with exactly n X's and no O's. Similarly, O_n is the number of rows, columns, or diagonals with just n O's. The utility function assigns $+1$ to any position with $X_3 = 1$ and -1 to any position with $O_3 = 1$. All other terminal positions have utility 0. For nonterminal positions, we use a linear evaluation function defined as $Eval(s) = 3X_2(s) + X_1(s) - (3O_2(s) + O_1(s))$.

(a) Approximately how many possible games of tic-tac-toe are there?

(b) Show the whole game tree starting from an empty board down to depth 2 (i.e., one X and O on the board), taking symmetry into account.

(c) Mark on your tree the evaluations of all the positions at depth 2.

(d) Using the minimax algorithm, mark on your tree the backed-up values for the positions at depths 1 and 0, and use those values to choose the best starting move.

(e) Circle the nodes at depth 2 that would not be evaluated if alpha-beta pruning were applied, assuming the nodes are generated in the optimal order for alpha-beta pruning.

Problem 2. Show that a single ternary constraint such as "$A + B = C$" cab be turned into three binary constraints by using an auxiliary variable. You may assume finite domains. (Hint: Consider a new variable that takes on value that are pairs of other values, and consider constraints such as "X is the first element of the pair Y.") Next, show how constraints with more than three variables can be treated similarly. Finally, show how unary constraints can be eliminated by altering the domains of variables. This completes the demonstration that any CSP can be transformed into a CSP with only binary constraints.

Problem 3 (Optional). Discuss how well the standard approach to game playing would apply to games such as tennis, pool, and croquet, which take place in a continuous physical state space.

Problem 4 (Optional). Describe how the minimax and alpha-beta algorithms change for two-player, non-zero-sum games in which each player has a distinct utility function and both utility functions are known to both players. If there are no constraints on the two terminal utilities, is it possible for any node to be pruned by alpha-beta? What if the player's utility functions on any state differ by at most a constant k, making the game almost cooperative?

Problem 5 (Optional). Give precise formulations for each of the following as constraint satisfaction problems:

(a) Rectilinear floor-planning: find non-overlapping places in a large rectangle for a number of smaller rectangles.

(b) Class scheduling: There is a fixed number of professors and classrooms, a list of classes to be offered, and a list of possible time slots for classes. Each professor has a set of classes that he or she can teach.

(c) Hamiltonian tour: given a network of cities connected by roads, choose an order to visit all cities in a country without repeating any.