CSC384 A3 Part I - Written Answers

Hunter Richards - 1002224158

2018-03-14

Question 1

1) A heuristic value for a game state is evaluated from a "score" which is some function of game features (i.e. number of food acquired, proximity to ghosts, number of capsules, actual in-game score, etc.). They also result in a commit to a move.

   A heuristic value for an A* search evaluates the cost/distance from the next state to the goal state. This acts as an attractive force between the goal and the search agent. Typically, low heuristic values are desired, indicating that the search agent is getting "warmer" as it searches for the goal state.

   Game search heuristics are "good" if they preserve the relative ordering of node scores, as dictated by the actual ordering of nodes determined from the in-game score. A* search heuristics are "good" if they are consistent, guarantee optimality, are monotone, and are admissible (typically). We want to know we have found an optimal path if we choose to use A* search (typically). Additionally, heuristics in both situations are "good" if they are efficient to compute.

Question 2

1) In minimax our evaluation function is purely the current score, Pacman seeks to maximize this. When death is inevitable and the score is decreasing with time, Pacman will commit suicide to obtain the highest score rather than trying to run away which will cost time and result in more points deducted.

2) a) Not same
   b) Not same
   c) Not same

Question 3

1) a) In the best case scenario, alpha-beta will prune at every opportunity and remove the children of all nodes except that of the left most nodes (assuming left-to-right traversal). Therefore, alpha-beta evaluates $b \times 1 \times b \times 1 \times \cdots \times b \times 1$ nodes, for a total time complexity of $O(b^{d/2})$. This implies that alpha-beta searches a depth of about $2d$, or twice that of minimax, in the same amount of time.
   b) In the worst case scenario, alpha-beta will never be able to prune and the tree remains intact. This reduces alpha-beta to simple minimax. Therefore, both minimax and alpha-beta will search a depth of d in the same amount of time. Theoretically, they have the same time complexity of $O(b^d)$. Practically, alpha-beta may perform slightly worse than minimax because it is making extra comparison computations in vain while minimax is not.

2) False, alpha-beta pruning only speeds up computation. It does not change the answer.