EECS 348 Introduction to Artificial Intelligence

Practice Midterm Exam

April 19, 2013

Partial credit will be given, so feel free to explain your work.

1. (10 points total) You have a map marking the top 10 restaurants in the country, and want to eat at all of them. You don't want to visit two steakhouses in a row, but otherwise simply want to minimize your total travel time.

   a) (6 points) Express your situation as a search problem. That is, define the states, operators, operator cost, and goal test you will use.

b) (1 point) What search algorithm would you use for this problem?

c) (3 points) Assume you're using A* search. Devise a heuristic that is admissible, and another that is inadmissible. With which will you obtain optimal solutions?

2. (10 points total) Consider the Tic-Tac-Toe board in Figure 0.1.

a) (6 points) X is to move. Draw the full minimax tree for this problem.

b) (2 points) According to the minimax algorithm, which move should the maximizing player make?

c) (2 point) Describe behavior for the O player in which the Minimax move given above is not optimal.

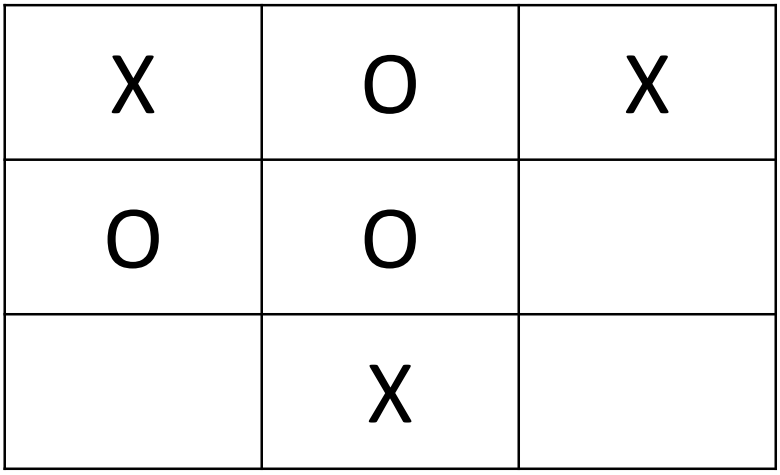

Figure 0.1: Minimax Example

3. (2 points) Briefly describe (1-2 sentences) two real-word search problems, one in which you would use Uninformed Search, and another in which you'd prefer to use A* search. In a sentence, state why the search technique you choose is preferable.

4. (2 points) In timed games, Minimax search is often used with iterative deepening depth-first search. In about three sentences, state why iterative deepening is preferable to simple depth-first search, or breadth-first search, over the Minimax tree.

5. (2 points) Consider a mad computer scientist who has a set of boolean variables $x_1, \ldots, x_n$ for n large. He secretly sets some subset of the variables to 1, and the rest to 0. Because he is mad, he demands that you guess which variables are set to 1. Because he is a computer scientist, he gives you an API that lets you check for a given guessed assignment of the n variables how many you got right. Describe in 2-3 sentences what kind of search

technique you'd use, and how many queries to the API you would require in the worst case.

6. (1 point) Is the search technique you described in the question directly above more similar to A* search, Constraint Satisfaction with Arc Consistency, or Local Search?