CS 4700 Fall 2019

Homework 2

Due: 18 October 2019, 2.40pm

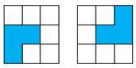
Consider the problem of taking an 2n x 3m checkerboard and "tiling" it with 2n x m "L trominoes" – pieces that fill 3 squares in the following shape  (each possibly rotated) – so that no piece extends over the edge of the checkerboard, no two pieces overlap, and collectively the pieces fully cover the checkerboard. Thus, for example, two L trominoes can cover a 2x3 checkerboard in the following two ways:  ,  .

1. Formulate this as a search problem. What are the states, operators, initial state, and goal condition? (Be more specific than, say, "Place a tromino" – think about what you would need to have specified for it to be clear how it might be mapped into a program.) Note that you can only place trominoes, not remove them or move them around.

2. For this part of the question, assume the search method used does not revisit states that it has already seen.
   a. Consider the case of a 2x3 board.
      i. Draw the entire search space for this problem. Do not create multiple copies of states that are identical. If there are multiple paths to the same state, show the multiple paths rather than showing an identical state in different parts of your diagram.
      ii. Show the order in which states would be visited by depth-first search using numbers 1, 2, 3, ….
      iii. Show the order in which states would be visited by breadth-first search using Roman numerals I, II, III, ….
      iv. Show the order in which states would be visited by iterative deepening search using letters A, B, C, …, Z, AA, AB, AC, ….
   b. Consider the case of a 3x3 board. (When relevant your answers should be a function of m and n.)
      i. As before, draw the entire search space. Notice that in this case there are states that are identical except for rotational or mirror symmetries. For example, consider the following two states,   , which are identical except for the orientation with which they appear on the page. Consider all states that can be made identical after suitable rotations and reflections to be the same state. In other words, when you check if you've previously visited a state consider it to be a match if the two are the same possibly after rotations/reflections, and if there are different paths to two states that are similarly a match consider them the same state with both paths leading to it.What is the branching factor for your state space?
   c. Consider the general 2nx3m case.
      i. Are there cycles in the search space? Is the search space a tree or a graph? Explain.
      ii. What is the branching factor for your state space?
      iii. What is the maximum depth the search can go?

iv. If the board can be covered by L trominoes, is the first terminal node reached by DFS guaranteed to be a solution? Explain.
v. When DFS halts, is it the case that every state in the space appears in either the open set or the closed set? Explain.

3. Consider again the general 2n x 3m case.
   a. For this part of the question, assume the search method used does *not* check for revisited states, so it might very well visit states multiple times, if relevant.
      i. Is depth-first search guaranteed to always halt? Explain.
      ii. Is depth-first search guaranteed to find a solution when one exists? Explain.
      iii. When breadth-first search halts, is it the case that every state in the space appears in either the open set or closed set? Explain.
   b. Imagine you were to use hill climbing search on this problem. Give a function f(s) such that the search is guaranteed to find a solution if it exists. Explain.
   c. Imagine using A* search on this problem.
      i. Give an admissible heuristic function h(s) for this problem. (Do not give the trivial answer for which h(s)=0 for all states.)
      ii. Given this h(s), is h'(s) = 2 x h(s) also admissible? Explain.

4. Consider a two-player game version of this task. Players alternate placing one L tromino on the board so that it covers three previously uncovered squares. The player to go last wins.
   a. Draw the game tree for this game for a 3x3 board in which you go first. Do not duplicate states that are identical except for reflections or rotations – treat them as the same state. (In other words, both of these should be treated as the same state:  .)
   b. Consider the case of a 4x4 board where your opponent went first in the center of the board as in the following diagram:  .
      i. Draw each of the successor nodes to just this state. As before, do not draw duplicate states.
      ii. Propose an evaluation function such that if I applied it to each of these states picking the one with the greatest value guarantees that I will win the game. It should not be specific to this orientation of the center piece – your function should work even if the board were, for example, rotated by 90 degrees.
   c. What is the worst-case branching factor for this game for a 2n x 3m board? Use big-O notation.

Just for fun, not part of the homework

- How many ways are there to tile a 2n x 3 checkerboard with L trominoes?
- Imagine you have an a x b chessboard, where ab = 3c+1 for some integer c. (In other words, the chessboard has one square more than a multiple of 3.) Is it always possible to remove one square from the chessboard so that the remaining squares can be covered with L trominoes?