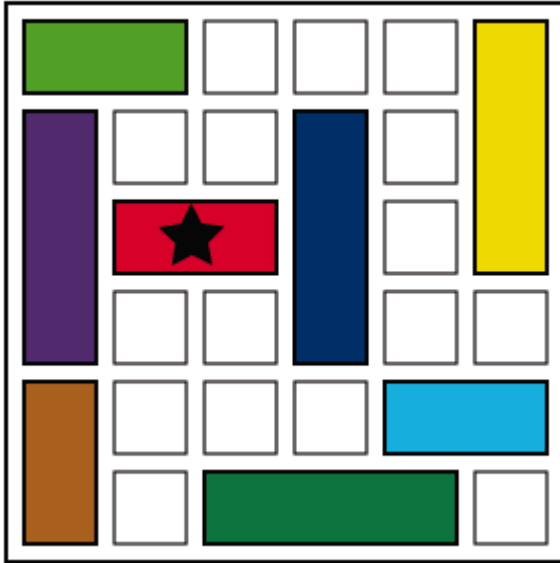


## Artificial Intelligence Problem Set 1

### Problem 1. (20 pts) Rush Hour

Consider the following "Rush Hour" puzzle. Recall that the rules for solving the puzzle are that vehicles can move only back or forward (no turning). The goal is to get the red car out of the parking lot through the exit at right.



2.1 (1 pt) Solve the problem and present your solution.

2.2 (7 pts) Give as accurate an estimate as you can for the size of the problem space for this puzzle. (The estimate should be for this instance of the puzzle, not a "general" estimate for an arbitrary Rush Hour puzzle.) Explain the rationale behind your estimate.

2.3 (7 pts) Suggest a plausible "h" function that would be appropriate for A\* search for a general instance of the Rush Hour puzzle. (The function should be a non-trivial but easily computable underestimate of the number of moves required to solve the puzzle from a given position.)

2.4 (5 pts) Did your solution in 2.1 have features in common with (e.g.) depth-first search? Breadth-first search? Means-ends analysis? Would bidirectional search be a good idea for this puzzle? Why or why not?

**Problem 2. (15 pts) Bidirectional Search**

Answer problem 3.15 (p. 116) in the Russell-Norvig text.

**Problem 3. (15 pts) Turing and Searle**

Choose your favorite of the "objections" from either the Turing or Searle papers and write a short paragraph or two explaining why you find that particular objection particularly provocative (regardless of whether you actually agree or disagree with it).

**Problem 4. (50 pts) Implementing Depth-First Search**

Consider the "milk-can transfer" problem shown in class. (Briefly: there are two 40-quart cans, both full; one empty five-quart can; and one empty four-quart can. Your job is to transfer milk between the cans without spilling until there are two quarts in each of the smaller cans.)

5.1 (5 pts) With as much accuracy as you can, calculate the size of the problem space for this problem. (Or, to put it another way: how many distinct achievable states are there?)

5.2 (45 pts) Write, in whatever language you prefer, a depth-first search algorithm that solves the milk-transfer puzzle and show a well-documented printout of your program and solution. Try to experiment with variations on the basic DFS algorithm shown in class: for instance, you might want to experiment with a version that maintains both an "open" and "closed" (i.e., "already-tested") list of nodes.