CS121 – Introduction to Artificial Intelligence – Spring 2008

Homework #4
(Search & Planning)
Out: 4/30/08 — Due: 5/8/08 (at noon)

How to complete this HW: First copy this file; then type your answers in the file immediately below each question; finally print this file and return it in the drawer marked CS121 of a file cabinet located in the entryway of the Gates building next to office Gates 182 (see red arrow in http://ai.stanford.edu/~latombe/cs121/2008/map.jpg) no later than Thursday 5/8 at noon.

Your name: ……………………………………………………………………………………….

Your email address: ………………………………………………………………………………

Note on Honor Code: You must NOT look at previously published solutions of any of these problems in preparing your answers. You may discuss these problems with other students in the class (in fact, you are encouraged to do so) and/or look into other documents (books, web sites), with the exception of published solutions, without taking any written or electronic notes. If you have discussed any of the problems with other students, indicate their name(s) here:
………………………………………………………………………………………………………
Any intentional transgression of these rules will be considered an honor code violation.

General information: Justify your answers, but keep explanations short and to the point. Excessive verbosity will be penalized. If you have any doubt on how to interpret a question, tell us in advance, so that we can help you understand the question, or tell us how you understand it in your returned solution.

Grading:

Problem#	Max. grade	Your grade
I	30	
II	15	
III	20	
IV	35	
Total	100	

I. Search (30 points)

You are given a roadmap of some country in the form of a connected non-directed graph in which nodes represent cities and edges represent roads between cities. (A connected graph is one in which every two nodes are connected by a path made of one or several consecutive edges.) Each edge (i,j) is labeled by the length $l(i,j)$ of the road between cities i and j.

Two friends live in two different cities, a and b, of the map. They want to meet in a city of the map (any one). To do this, they move in successive turns. On every turn, the two friends start moving at the same time. Each friend moves to a neighboring city on the map; he/she cannot stay in the same city. The amount of time needed to move from city i to neighboring city j is equal to the length $l(i,j)$ of the road between cities i and j. So, the two friends may not reach their respective new cities at the same time. The friend that arrives first to his/her new city must wait until the other arrives to his/her new city (each one calls the other on his/her cell phone when he/she arrives to a new city) before the next turn can begin. The two friends want to meet as quickly as possible. Note that the goal for the two friends is to meet in a city, not anywhere on a road.

1. Formulate this problem as a state-space search problem:
   a) What is the state space?
   b) What are the initial and the goal states?
   c) What is the successor function?
   d) What is the step cost function?

2. Let $D(i,j)$ be the straight-line distance between any two cities i and j in the map. Which, if any, of the following heuristic functions are admissible? Why?
   a) $D(i,j)$
   b) $D(i,j)-2$
   c) $D(i,j)/2$

3. Is the following statement true or false: “There are connected maps for which no solution exists”? If you answer ‘true’, give an example of such a map. If you answer is ‘false’, prove it.

II. Sampling Strategy in PRM Planning (15 points)

Consider the following sampling strategy to generate each node (milestone) of a probabilistic roadmap:

I. Pick a configuration ***q*** uniformly at random in the configuration space
II. If ***q*** tests collision-free then retain ***q*** as a node of the roadmap with very small probability (say, 0.01)
III. Else
   a. Pick a configuration ***q’*** uniformly at random in a small ball around ***q***
   b. If ***q’*** is collision-free then retain it as a node of the roadmap with small probability (say, 0.1)
   c. Else
      i. Let ***q”*** be the midpoint between ***q*** and ***q’***
      ii. If ***q”*** is collision-free then retain it as a node of the roadmap with probability 1

The strategy performs steps I, II, and III repeatedly to generate several hundred nodes.

1. What is the effect of this strategy on the distribution of the nodes of the roadmap? Can you more specifically explain what Step III.c tries to do? [Hint: Think of narrow passages.] Give an approximate representation of the distribution of nodes that would be generated by this strategy on the two-dimensional configuration space of Figure 1, where the darker region is the forbidden region and the feasible region is shown in white [Here, we simply ask you to distribute a collection of points drawn by hand; put more points in regions where the strategy will sample more configurations. The numerical probabilities 0.01 and 0.1 at Steps II and III.b are indicative only. Don't take them literally,]:

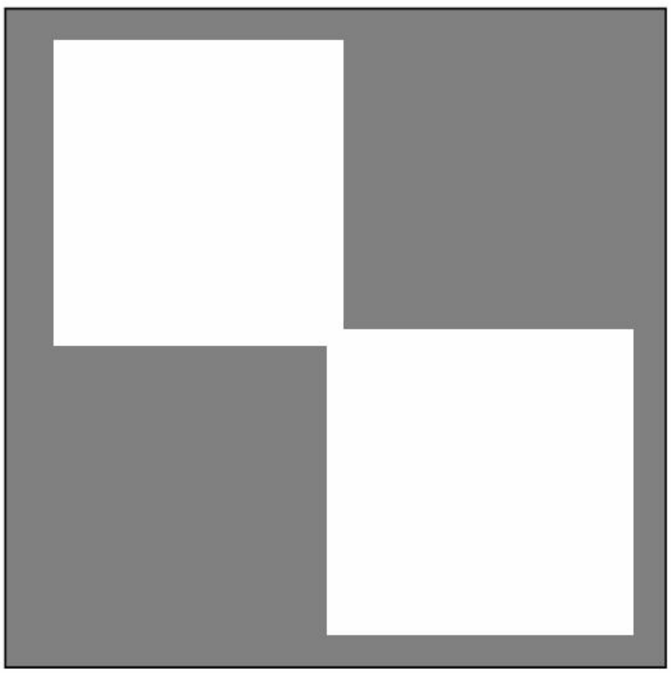

Figure 1: Two-dimensional configuration space for Question II.1.

2. [In this question we consider Step III.c again. The question is no longer about what this step does, but about why it can lead to a more efficient motion planner.] The generation of a node at step III.c.ii requires that three configurations be tested for collision. In addition, this step has little chance to generate a roadmap node. So, Step III.c might look like a waste of computational time. Why can it be nevertheless a good idea? [Hint: Testing a path between two nodes for collision is much more expensive in general than testing a single configuration.]

III. Motion Planning: Configuration Space (20 points)

Consider the two-link/two-joint robot arm shown in Figure 2. The first link rotates about point J_1 (which is fixed); the second link rotates about point J_2 (which moves with link 1). The orientation of the first link is defined by the angle $\theta_1 \in [0,2\pi]$ and the orientation of the second link is defined relative to the first link by the angle $\theta_2 \in [0,2\pi]$. Both angles are measured as shown in Figure 2.

The black regions (a square box and a vertical wall) depict obstacles. The initial configuration of the robot is in plain line and its goal configuration is shown in thick dotted line.

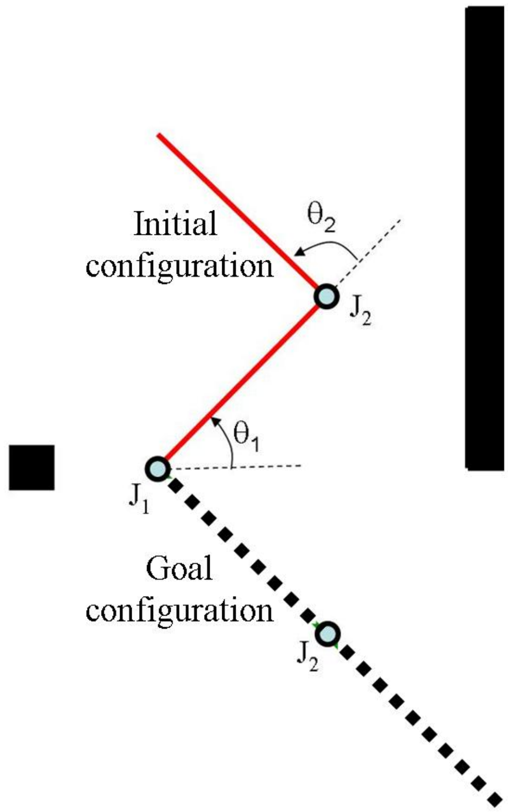

Figure 2: Robot arm and obstacles

1. The 4 drawings in Figure 3 present four copies of the configuration space $[0,2\pi]\times[0,2\pi]$ of this robot. In each, the horizontal axis corresponds to θ_1 and the vertical axis to θ_2, as shown in drawing (a); the dark regions are candidate maps of the obstacles. Which drawing depicts the correct map: (a), (b), (c), or (d)? [Here, we do not ask you to check the details of the drawings. Very simple observations should allow you to eliminate three of drawings.]

2. In the drawing you have selected, mark the start configuration with a plus (+) and the goal configuration with a bold bullet (•). Draw a collision-free path for the robot to move from the initial configuration to the goal configuration.

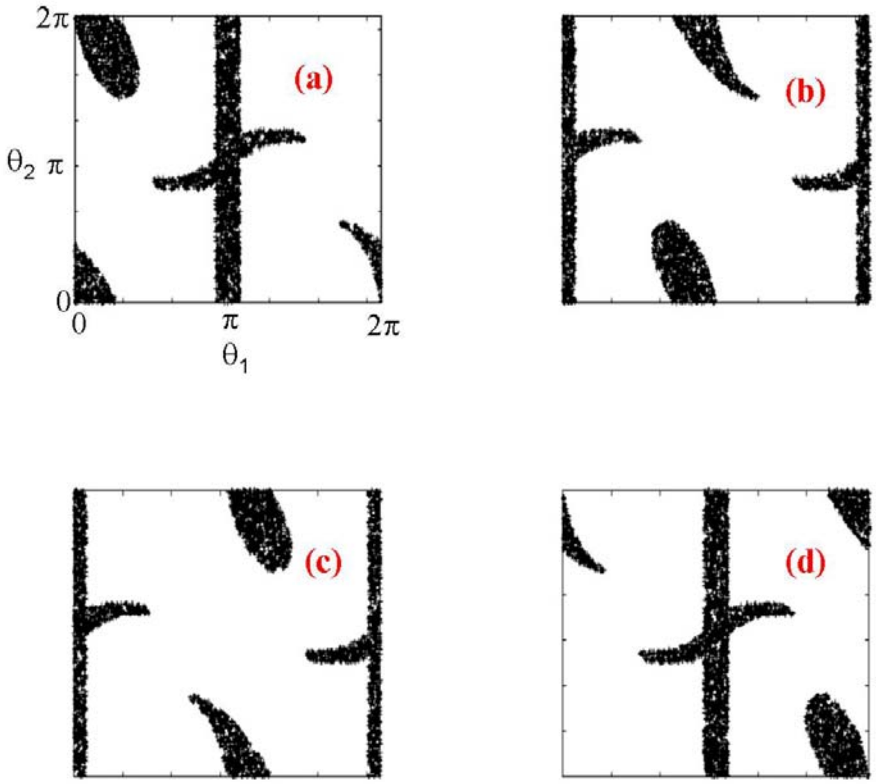

Figure 3: Potential configuration spaces for Question III.1

IV. Action Planning: Riding an Elevator (35 points)

A robot is on the 2nd floor of a building and wishes to go to the 5th floor. The initial state is described by the following conjunction of propositions:

$$\text{ON(Robot,2)} \wedge \text{WORKING(Elevator3)} \wedge \text{ON(Elevator3,8)}.$$

The goal of the robot is defined by: ON(Robot,5).

The action of the robot calling an elevator e from floor f_1, so that the elevator eventually arrives at f_1, is described by the following STRIPS action schema:

Call(e,f_1)
P = ON(Robot, f_1) $\wedge$ ON(e, f_2) $\wedge$ WORKING(e)
D = ON(e, f_2)
A = ON(e, f_1)

Note that the location of the elevator is updated.

1. Write the STRIPS action schema **Ride**(e,f_1,f_2) that represents the action of the robot riding an elevator e from floor f_1 to floor f_2.

2. Is **Call**(Elevator3,5) relevant to achieving ON(Robot,5)? Is **Ride**(Elevator3,2,5) relevant to achieving ON(Robot,5)?

3. Express in English what the regression of a goal through a STRIPS action is. Is it a condition, or an action, or something else? If it is a condition, what is its meaning?

4. Compute the regression of the goal ON(Robot,5) through **Ride**(Elevator3,2,5).

5. Compute the regression of the goal ON(Robot,5)$\wedge$ON(Elevator3,4) through **Ride**(Elevator3,2,5)? Is the obtained condition achievable? If not, why can't the planner detect that immediately? What additional knowledge would have to be given to the planner to make this detection possible? Is this knowledge implicit in the descriptions of **Ride** and **Call**?

6. Let the initial state be the one given at the beginning of this problem and the goal of the robot be ON(Robot,5). Let the robot's planner be a breadth-first backward planner. So, the root of the search tree will be labeled by ON(Robot,5). The arcs of the tree will be labeled by actions and the other nodes will be labeled by regressed conditions. When level 2 is completed, there will be a path in the tree whose first arc is labeled by **Ride**(Elevator3,2,5) and second arc is labeled by **Call**(Elevator3,2). What is the regressed condition labeling the node at the end of this path? Is this condition satisfied in the initial state? How?