

Homework #4
(Motion and Task Planning)
Out: 1/31/11 — Due: 2/8/11 (at noon)

How to complete this HW: First copy this file; then type your answers in the file immediately below each question; start each question on a separate page, write your name on every page. 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/~lalombe/cs121/2011/map.jpg>) no later than Tuesday 2/8 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	25	
II	25	
III	25	
IV	25	
Total	100	

Name

I. Motion Planning: Coordination of two robots

In Figure 1 two square robots A and B move in a 2D workspace. Each side of a robot has length equal to 4 units. The robots cannot rotate. Each moves at fixed orientation along a distinct track so that its center remains on a bold line shown in the figure. The robots must not collide with each other.

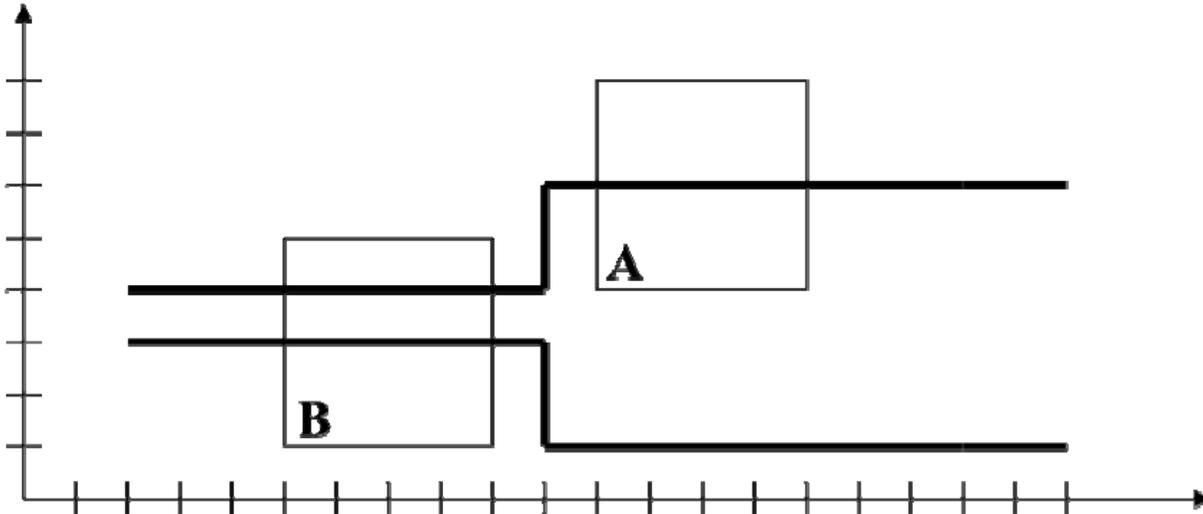


Figure 1: Two-robot system

Even though there are two robots, we can still represent both of them as a point in a 2D configuration space. To help you, we tell you that this configuration space is the 20x20 square shown in Figure 2. The horizontal axis corresponds to the position of robot A along its track and the vertical axis corresponds to the position of robot B. So, a point in this 2D configuration space uniquely defines the positions of the two robots. Assume that (0,0) would place each of the two robots at its leftmost position on its track. Similarly (20,20) would place each robot at its rightmost position.

The grid shown in Figure 2 is intended to help you answer the two questions of this problem precisely. Each small square of this grid has size 1 unit x 1 unit. The unit of length in Figure 2 is not shown at exactly the same scale as the unit of length in Figure 1.

1. In Figure 2, mark by a thick dot the point representing the configuration of the robots shown in Figure 1.
2. In Figure 2, draw the forbidden region where the two robots collide or touch each other.

Name

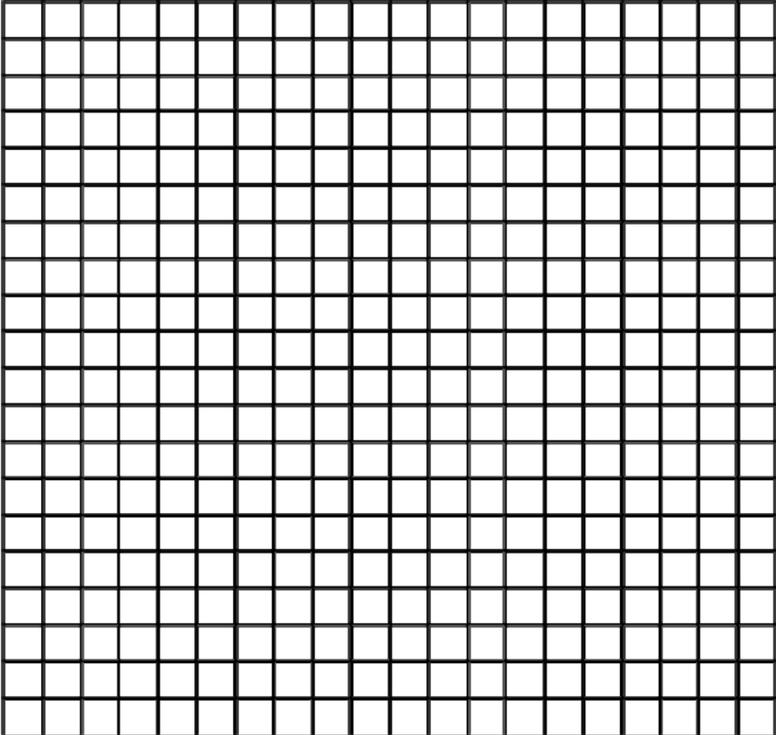


Figure 2: 2D configuration space of the two-robot system

II. Multi-Step Motion Planning

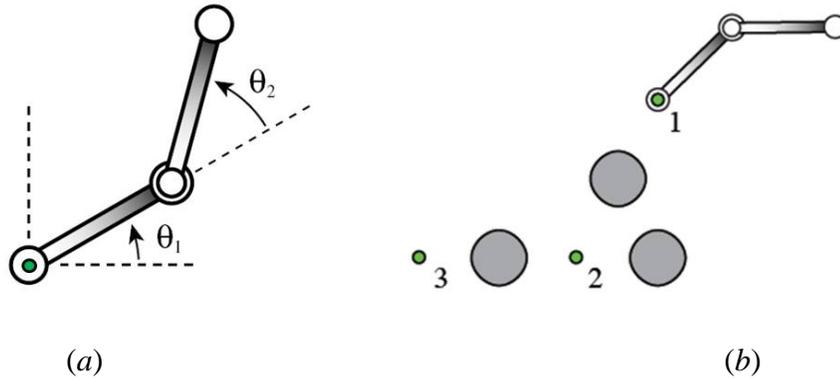


Figure 3: The robot R considered in Problem II and its workspace

In this problem we consider the planar robot R depicted in Figure 3(a). It consists of two identical rods connected by a revolute joint (in the middle). The environment of the robot is shown in Figure 3(b). It contains three obstacles and three pegs. The obstacles are the 3 large discs shown gray. The pegs are the 3 small circles labeled 1 to 3. The pegs are not obstacles for the robot. In Figure 3(b) the robot R is attached to peg 1 at one of its extremities. One of the two rods is “above” the other, so that it can cross over the other without collision.

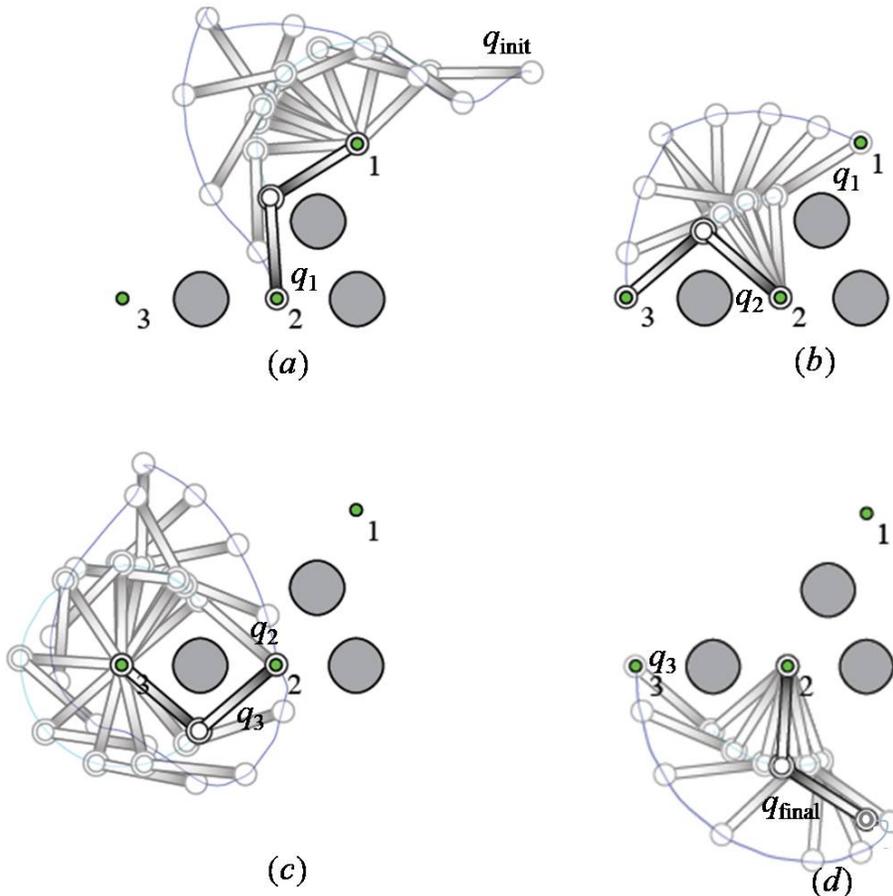


Figure 4: A possible motion sequence of robot R

Name

At any one time R must have one of its two extremities attached at a peg position around which it can rotate. For instance, in Figure 3(b), one end of R is attached at peg 1. In this case, the rod attached to peg 1 can rotate around peg 1. The other rod can simultaneously rotate around the revolute joint in the middle of R . One such motion is shown in Figure 4(a). In this motion, R moves from the configuration q_{init} shown in Figure 3(b) to a configuration q_1 where it reaches peg 2 with its other extremity. At the end of this motion, R can attach itself to peg 2 and detach from peg 1 (though one extremity remains at peg 1). Then, R can perform the motion in Figure 4(b) from configuration q_1 to configuration q_2 where it reaches peg 3 with its moving extremity. There R can attach itself to peg 3 and detach from peg 2. Figure 4(c) and (d) show two more motions of R , from q_2 to q_3 , and from q_3 to q_{final} . In each figure, the configurations along the motion are shown in shaded graphics, except the last one. The continuous thin line shows the path traced out by the moving extremity of R .

1. To move R must be attached to a single peg. The configuration of R is then defined by two angles (θ_1, θ_2) , where θ_1 is the angle made by the rod attached to the peg with the horizontal axis and θ_2 is the angle made by the second rod with the first rod (see Figure 3(a) for an illustration). Each of the two angles can vary between $-\pi$ and $+\pi$. Angles are counted positively in the counterclockwise direction and negatively in the other direction. If an angle reaches $-\pi$ (resp. $+\pi$) in one direction, it switches to $+\pi$ (resp. $-\pi$), meaning that the robot is not limited in its rotations, except possibly by the obstacles.

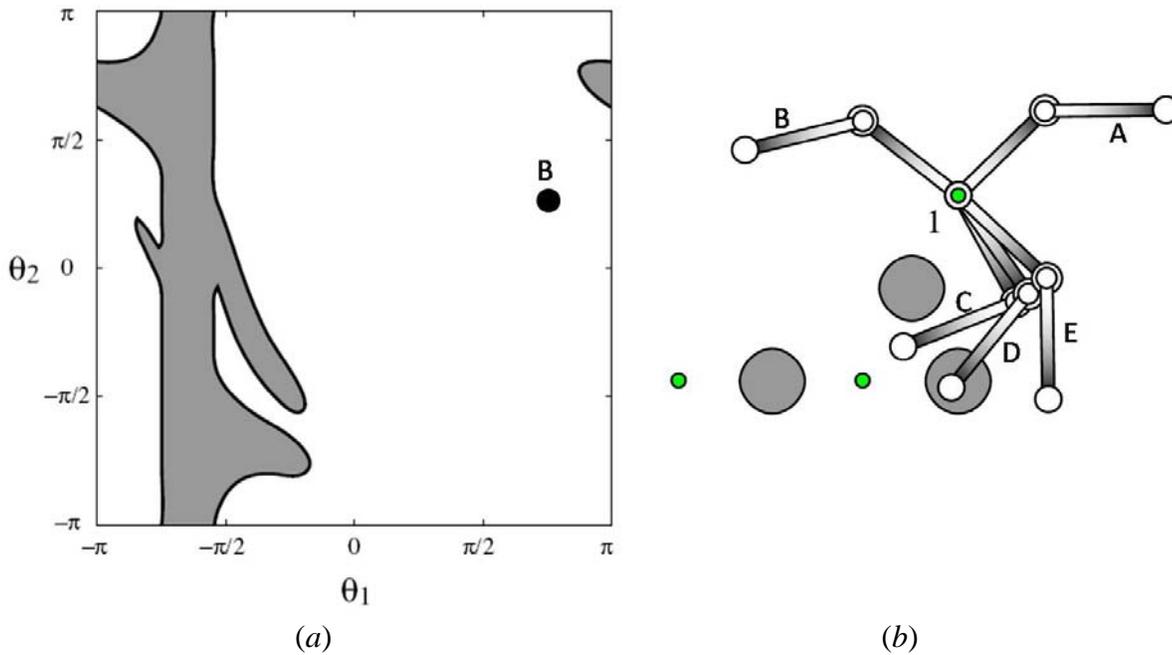


Figure 5: Configuration space of R at peg 1

- (i) Figure 5(a) shows the configuration space of R when it is attached to peg 1. The gray region is the region where the robot collides with an obstacle. Represent each of the 5 configurations labeled A, C, D, and E in Figure 5(b) by a bold point in Figure 5(a). Besides each of the 5 point indicate clearly the corresponding configuration (A, C, D, or E). To help you we have already mapped configuration labeled B in this space.

Name

(ii) The collision-free region for the robot is shown white in Figure 5(a). A connected component X of this region is such that any two configurations in X can be connected by a collision-free path. How many connected components does the collision-free region shown in Figure 5(a) contain? [Hint: Remember that $(\theta_1, -\pi)$ and $(\theta_1, +\pi)$ are the same configuration; so are $(-\pi, \theta_2)$ and $(-\pi, \theta_2)$.]

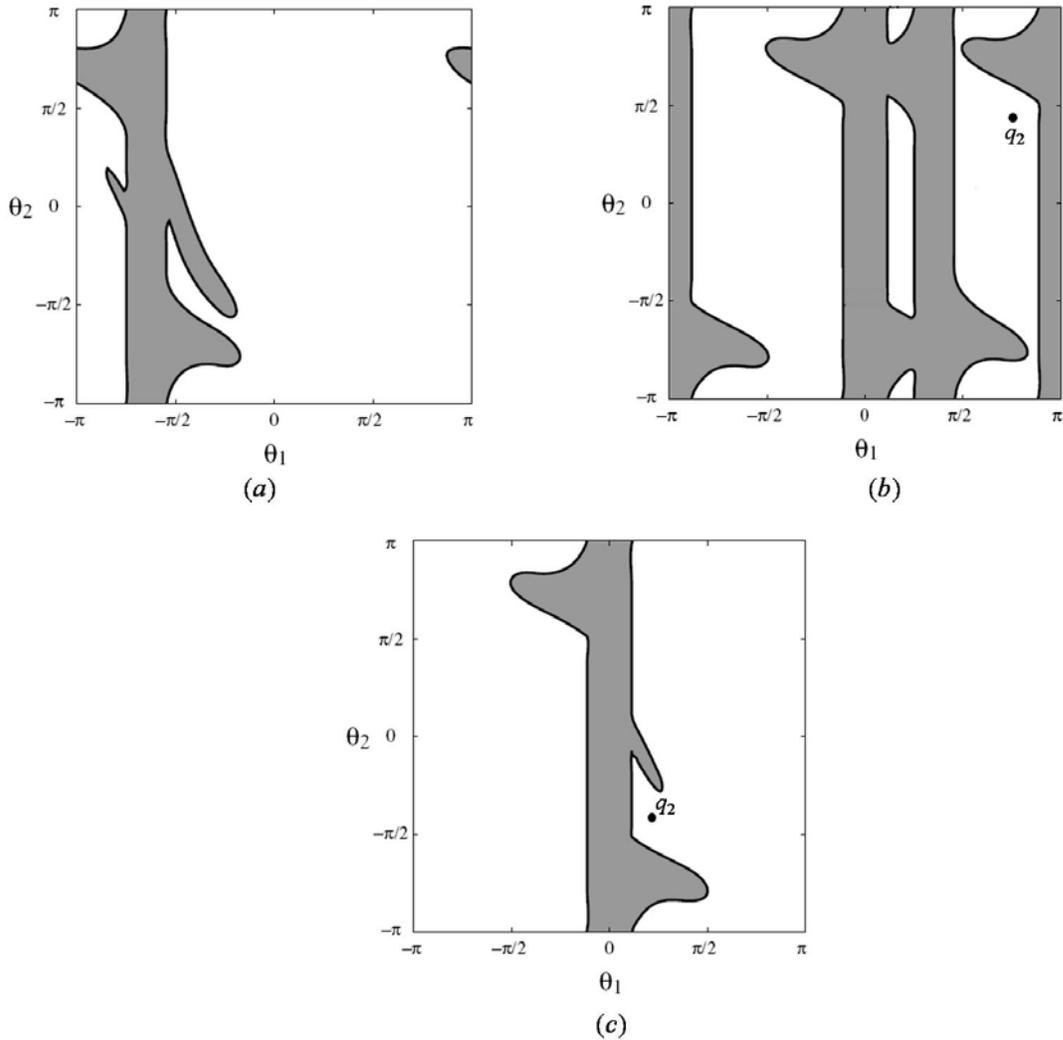


Figure 6: The configuration spaces of R when it is attached to peg 1 (Figure (a)), to peg 2 (b), and to peg 3 (c)

2. Figures 6(a), (b), and (c) show the configuration spaces of R when it is attached to pegs 1, 2, and 3, respectively. In these configuration spaces draw (approximately) the successive paths followed by R when it performs the motion sequence shown in the 4 drawings of Figure 4. To do this, proceed as follows:

(i) Let q_{init} be the initial configuration in Figure 4(a), also shown in Figure 3(b). Let q_1 , q_2 , and q_3 be the configurations reached by the motions shown in Figure 4(a), (b), and (c), respectively. Let q_{final} be the configuration reached by the motion shown in Figure 4(d). Mark each of these configurations by a bold point in Figure 6. Indicate the name of the

Name

configuration beside each point. [Hints: Every configuration, except q_{init} , maps to two points in two different spaces. In addition, when the robot switches between two pegs (i.e., detach from one peg to attach at the other peg, the fixed extremity changes also, so does the definition of (θ_1, θ_2) . To help you, in Figure 6 we have shown q_2 in the two spaces where R is attached to peg 2 and to peg 3.]

- (ii) Now, in Figure 6, connect q_{initial} to q_1 by a path in the appropriate space, q_1 to q_2 in the appropriate space, ..., and q_3 to q_{final} in the appropriate space. [Hints: Some of the paths may cut the lines $\theta_1 = \pm\pi$ and $\theta_2 = \pm\pi$, and so may consist of multiple curve segments. Note that two out of the 4 paths are in the same space.] Your paths need not be accurate (in fact, this would be quasi-impossible to do by hand), but they must cross the lines $\theta_1 = \pm\pi$ and $\theta_2 = \pm\pi$ whenever needed and only when needed (the positions of the crossing points do not have to be accurate either).
3. How many connected components does the collision-free region shown in Figure 6(b) contain?
4. The Probabilistic Roadmap (PRM) approach was presented in class for a robot with one configuration space. But here R has several configuration spaces, one for each peg where it can attach itself. Describe briefly how you would extend the PRM approach to automatically plan a collision-free path of R between two given configurations. [Note: We do not ask you to re-describe the PRM approach presented in class. We only ask you to describe how you would extend the approach for R .]

Name

III. Sampling Strategy in PRM Planning

1. Consider the following sampling strategy, SAMPLE#1, to generate each node (milestone) of a probabilistic roadmap:

SAMPLE#1:

- A. Pick a configuration q uniformly at random in the configuration space
- B. If q tests collision-free then retain q as a node of the roadmap with small probability
- C. Else
 - a. Pick a configuration q' uniformly at random in a small ball around q
 - b. If q' is collision-free then retain q' as a node of the roadmap with probability 1

This strategy is applied repeatedly to generate N nodes, where N ranges typically in the thousands.

What is the effect of this strategy on the distribution of the nodes of the roadmap? Can you more specifically explain what Step C tries to do? Give an approximate representation of the distribution of nodes that would be generated by this strategy on the two two-dimensional configuration spaces of Figure 7(a) and 7(b), where the gray 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 in the two spaces of Figure 7. Put more points in regions where the strategy will sample more configurations.]

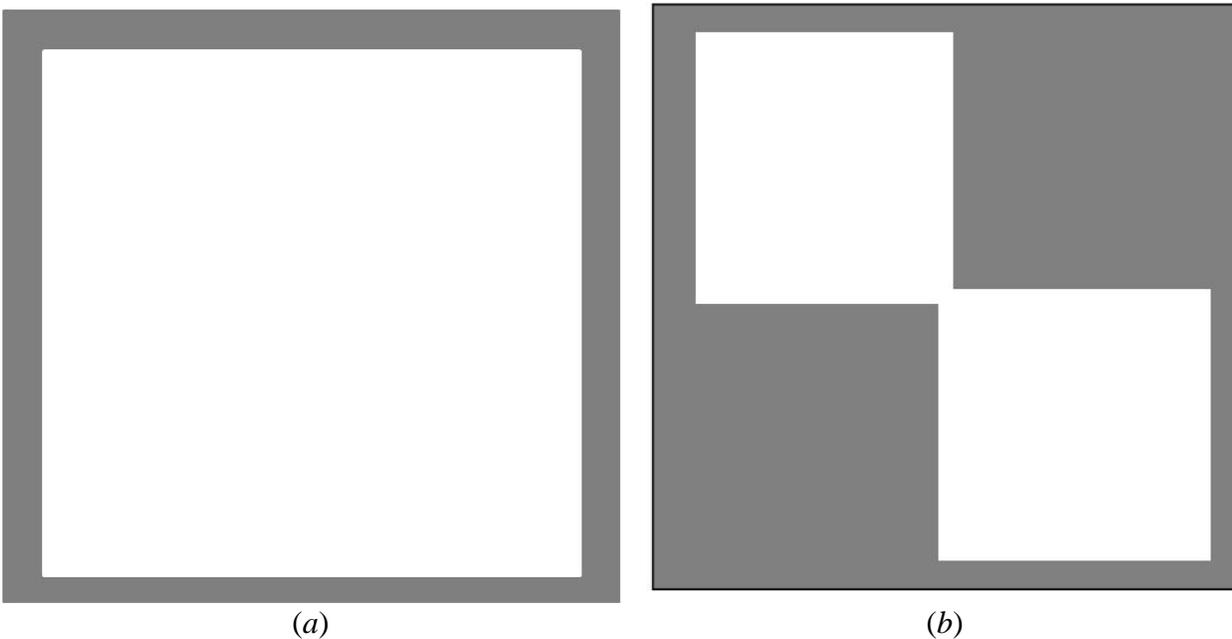


Figure 7: Two two-dimensional configuration spaces for Question III.1.

2. Consider now the following sampling strategy, SAMPLE#2:

SAMPLE#2:

- A. Pick a configuration q uniformly at random in the configuration space
- B. If q tests collision-free then retain q as a node of the roadmap with very small probability p_1

Name

- C. 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 p_2 [where p_2 is small, but much larger than p_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

What is the effect of this strategy on the distribution of the nodes of the roadmap? Explain what Step C.c tries to do. Give an approximate representation of the distribution of nodes that would be generated by this strategy on the two two-dimensional configuration spaces of Figure 8(a) and 8(b).

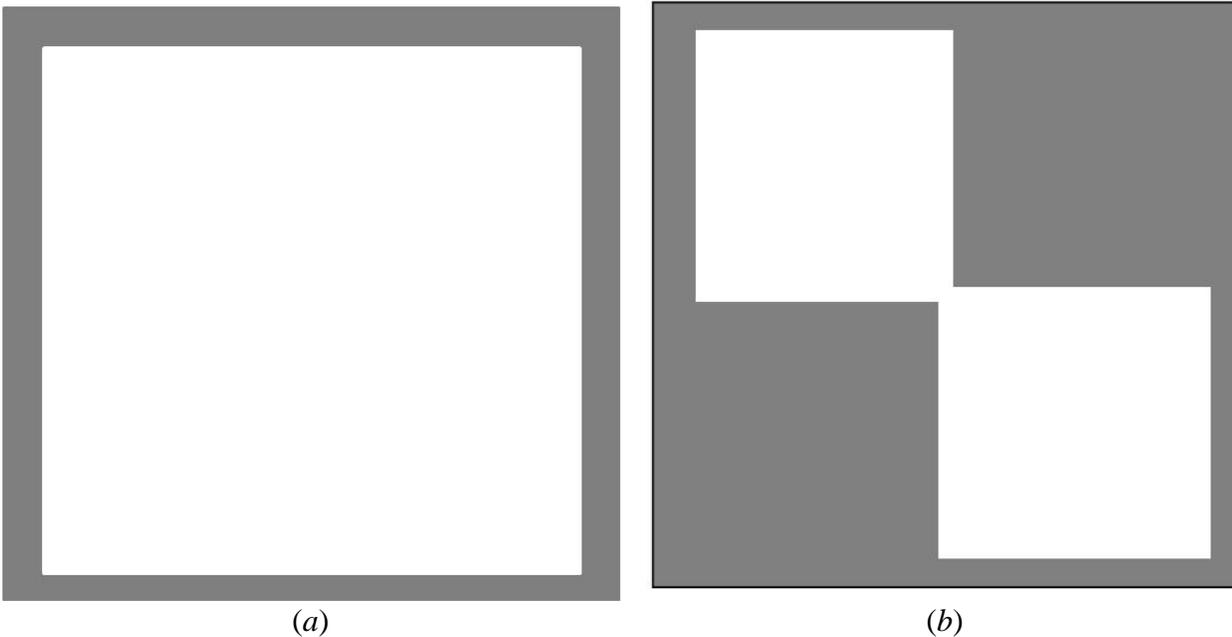


Figure 8: Two two-dimensional configuration spaces for Question III.2.

- 3. [This question is about Step C again. It is no longer about what this step does, but about why it can lead to a more efficient motion planner.] The generation of a roadmap node by SAMPLE#2 requires that up to three configurations be eventually sampled and tested for collision. Moreover, SAMPLE#2 has less chance to eventually generate a roadmap node than a mere uniform random sampling strategy that would keep any configuration sampled in the feasible space. Nevertheless, in practice, a probabilistic roadmap planner using SAMPLE#2 is often much faster than a planner using a uniform random sampling strategy.

We tell you that on average testing a connection between two roadmap nodes for collision is much more expensive than testing a single configuration (say, 10 times more expensive). Can you explain why SAMPLE#2 may be a good idea? [Your answer does not have to be quantitative. In particular, you don't have to write any kind of formula describing running times. The expression "10 times more expensive" should not be taken literally; our intention is only to give an order of magnitude of what we mean by "much more expensive".]

Name

IV. Action Planning

We consider the famous Christmas-Candy problem. In a room, there is a child and a Christmas tree. Some candy is hanging from the tree. The child is not tall enough to reach the candy, but there is a chair in the room that allows the child to grab the candy if he/she climbs on it.

Initially the child is at position A, the Christmas tree is at position B (and the candy is at the same position), and the chair is at position C. All three positions are in the same room. The child, the chair and the tree are on the floor. The candy is high.

The only actions available to the child are:

- $GO(x,y)$: where the child goes from position x to position y in the room
- $PUSH(x, y, z)$: where the child pushes x from position y to position z in the room
- $CLIMB(x)$: where the child climbs onto the chair at position x
- $GRASP(x, y)$: where the child grasps x at position y

To perform the action GO , the child must be on the floor. Under this condition, GO allows the child to go from any position to any other position in the room with a single action. The action $GRASP(x,y)$ allows the child to hold x if x and the child are both at position y , x is high, and the child is on the chair. The child can only hold one object at a time. The child does not have to be holding an object in order to $PUSH$ it. In order to $PUSH$ an object, both the object and the child must be on the floor at the same position. While the child performs $PUSH$, the child changes position with the object being pushed, with both being at the same position at each time.

Using the STRIPS language, we express the initial state of the planning problem by the following set (conjunction) of eight propositions:

$$AT(\text{Child}, A) \wedge AT(\text{Candy}, B) \wedge AT(\text{Chair}, C) \wedge ON\text{-FLOOR}(\text{Child}) \wedge ON\text{-FLOOR}(\text{Chair}) \\ \wedge ON\text{-FLOOR}(\text{Tree}) \wedge HIGH(\text{Candy}) \wedge HANDS\text{-FREE}(\text{Child})$$

In the following, we will also need other similar propositions like $AT(\text{Child}, B)$, $AT(\text{Child}, C)$, $AT(\text{Chair}, B)$, $ON\text{-CHAIR}(\text{Child})$, as well as the proposition $HOLDING(\text{Child}, \text{Candy})$.

A goal state is any state that contains $HOLDING(\text{Child}, \text{Candy})$.

1. Write down the STRIPS representations (schemas) for the actions $GO(x,y)$, $CLIMB(x)$, and $GRASP(x,y)$ that the child may perform. That is, provide the precondition and delete and add lists for each action.

To help you, we give you the representation of $PUSH(x,y,z)$:

$PUSH(x, y, z)$:

P: $AT(\text{Child}, y) \wedge AT(x, y) \wedge ON\text{-FLOOR}(\text{Child}) \wedge ON\text{-FLOOR}(x) \wedge (x \neq \text{Child})$

D: $AT(\text{Child}, y), AT(x, y)$

A: $AT(\text{Child}, z), AT(x, z)$

Write down the representation of the three other actions.

2. List all the actions whose preconditions are satisfied in the initial state (there should be no variables in these actions). No justification is required here.
3. Compute the regression of $HOLDING(\text{Child}, \text{Candy}) \wedge AT(\text{Chair}, B)$ through $GRASP(\text{Candy}, B)$. Just give the result. You don't have to explain how you do the computation.