Robotics and Motion Planning — Spring 2002 — Dan Halperin

Assignment no. 2

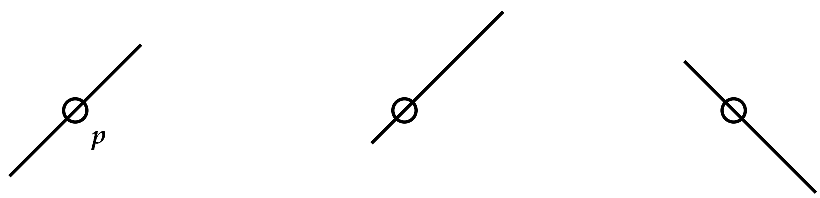
http://www.cs.tau.ac.il/~danha/courses/rob02.html

due: May 7th, 2002

You may submit this assignment as well in pairs.

Exercise 2.1 **(a)** What is the maximum combinatorial complexity (give asymptotic lower and upper bounds) of the free space of the following planar motion-planning problems. In both cases the obstacles are pairwise interior-disjoint polygons with a total of n vertices.

- An anchored 2-link arm consisting of two segments and two rotational degrees of freedom (namely the first two links of the 3-link arm of Assignment 1).
- A robot arm with two degrees of freedom consisting of a line segment that passes through a point p in the plane. It can rotate around p and translate through p, but at all times it coincides with p. See the following figure for an illustration.



(b) Devise efficient algorithms to compute the free space in both cases.

Exercise 2.2 We denote the Minkowski sum operation by $\oplus$. Let P_1 and P_2 be two convex polygons. Let S_i denote the set of vertices of P_i. Prove that

$$P_1 \oplus P_2 = \text{ConvexHull}(S_1 \oplus S_2)$$

Exercise 2.3 In class we showed an $O(n)$ bound on the complexity of the union of a set of polygonal pseudo-disks with n vertices in total. In this exercise we are concerned with the precise bound.
(a) Assume that the union boundary contains m original vertices of the polygons. Show that the complexity of the union boundary is at most $2n - m$. Use this to prove an upper bound of $2n - 3$ on the complexity of the union boundary.
(b) Prove a lower bound of $2n - 6$ by constructing an example that has this complexity.

There are additional **exercises** on the other side of the page.

Exercise 2.4 (p) A central component in many practical motion planners is a data structure that efficiently answers *collision detection* queries: Given the robot in a fixed configuration, determine whether it intersects the obstacles.

We wish to devise a collision detector for the following setting: the robot is a vertical line segment S of length l that translates in the plane, and the obstacle is a convex polygon P given by the list of its vertices in counterclockwise order.

A query to the detector is specified by the location (coordinates) of the lower endpoint of S. All the information (the length l, the coordinates of the vertices of P and of the query) will be given as integers (that is, the input integers fit in standard machine word [4 byte] integers). The answer should be *exact*, and therefore has to be computed exactly, not allowing numerical imprecision.

(a) one query Write an efficient procedure detect_collision that gets a query point q and returns 1 when the robot is intersecting the interior of P, 0 if the robot touches the boundary of P but not its interior, and -1 when the robot is in a free configuration. The procedure also gets a preprocessed data structure to answer the query efficiently.

(b) batch (bonus) Write a procedure batch_collision_detection that gets a number m and m queries in a vector query, and outputs an integer vector answer, where answer[i] is the answer to the collision-detection query[i].

Your procedures should comply with the prototypes given in the TA's site so that they can be tested automatically.

Exercise 2.5 We are given a convex polygonal robot P with m vertices that is free to translate inside a convex polygonal room Q with n vertices. The only obstacles to the motion of P are Q's walls. What is the maximum combinatorial complexity of the free space in this case? Describe an efficient algorithm to compute it.