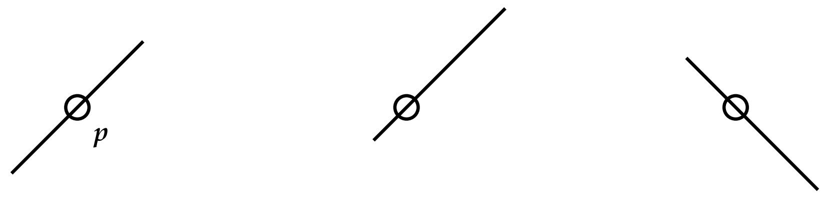
Assignment no. 2

due: Monday, December 2nd, 2019

Exercise 2.1 A Roomba is moving in a room with convex polygonal obstacles, which are pairwise interior-disjoint, with a total of n vertices.
(a) Show that the set of C-obstacles (namely configurations-space obstacles, or expanded obstacles) each corresponding to one convex polygonal obstacle, is a set of pseudo-discs.
(b) optional (bonus) Prove that the complexity of the free space in this case is $O(n)$. Do not rely on the general result for pseudo-discs; however, you can use the results that we have seen for the special cases of discs and/or of polygons.

Exercise 2.2 **(a)** What is the maximum combinatorial complexity (give asymptotic lower and upper bounds) of the free space of the following planar motion-planning problem. A robot arm with two degrees of freedom moving in the plane among polygonal obstacles with a total of n vertices. The arm consists 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 an efficient algorithm to compute the free space.

Exercise 2.3 (p2) Write a program that solves the following motion-planning problem. We are given a simple polygon A, the robot, which can translate in the plane, and a set of pairwise interior-disjoint simple polygons, the obstacles, which the robot has to avoid. We are also given a pair of planar points s and g denoting the reference point of A at a start position and a desired goal position respectively. The program decides whether a *semi-free path*[1] for A from start to goal exists and if so outputs such a path.

In the course's webpage you will find ample helpful information for solving this problem, as well as the input/output format. You will also find there is a Python template for writing the program, together with a graphical interface to display your solution path. You do not have to use the given template; you can produce a solution by your own program and still use the graphical verification tool that is provided, by uploading your solution to a file.

Exercise 2.4 is on the next page.

[1]Semi-free path means that the robot is allowed to touch the obstacles, but not to penetrate into them.

Exercise 2.4 (p2), optional (bonus) Same as Exercise 2.3, only this time you are required to produce a *good* path. Choose your criterion: length, clearance, smoothness, etc. and devise a path that is good according to this criterion; not necessarily optimal, but better than an arbitrary path that the method that puts a point inside each trapezoid and on each vertical edge produces.