Algorithmic Robotics and Motion Planning

Spring 2011

Motion planning and arrangements

Dan Halperin

School of Computer Science

Tel Aviv University

Today's lesson

- we will learn a general framework for complete modeling of motion planning (and other) problems
- we will then apply it to solve two systems with two degrees of freedom in the plane:
  - a disc moving among disc obstacles
  - an L-shaped robot translating among polygonal obstacles
- we will review general complexity results

Motion planning: the basic problem (reminder)

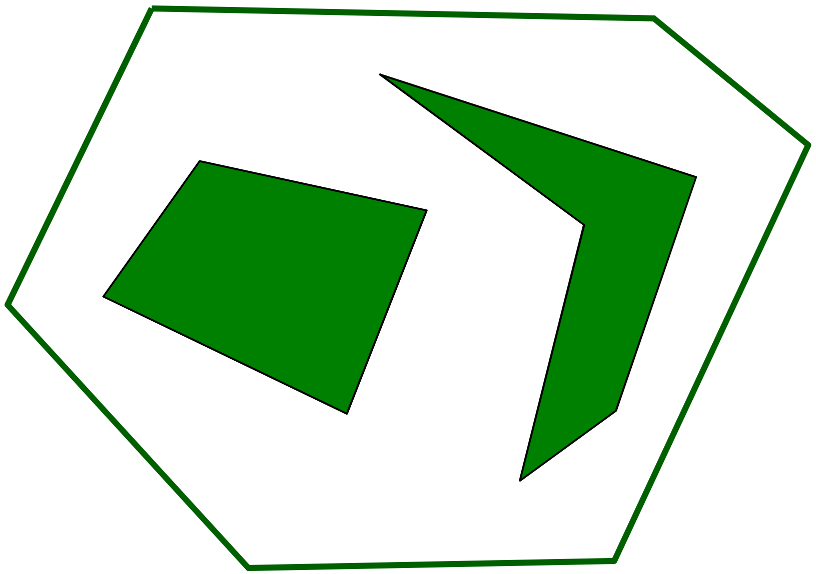
Let B be a system (the robot) with k degrees of freedom moving in a known environment cluttered with obstacles. Given free start and goal placements for B decide whether there is a collision free motion for B from start to goal and if so plan such a motion.

Configuration space

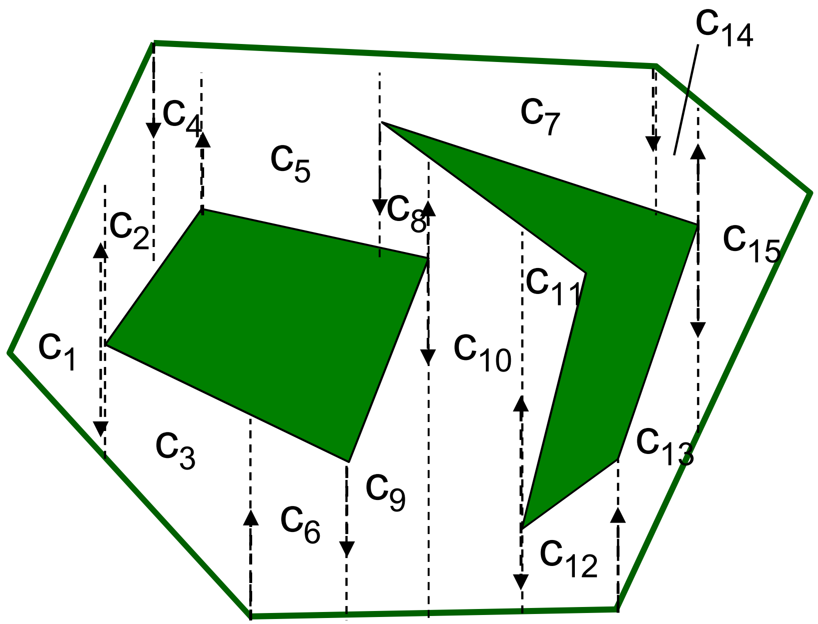
of a robot system with k degrees of freedom

- the space of parametric representation of all possible robot configurations
- C-obstacles: the expanded obstacles
- the robot -> a point
- k dimensional space
- point in configuration space: free, forbidden, semi-free
- path -> curve

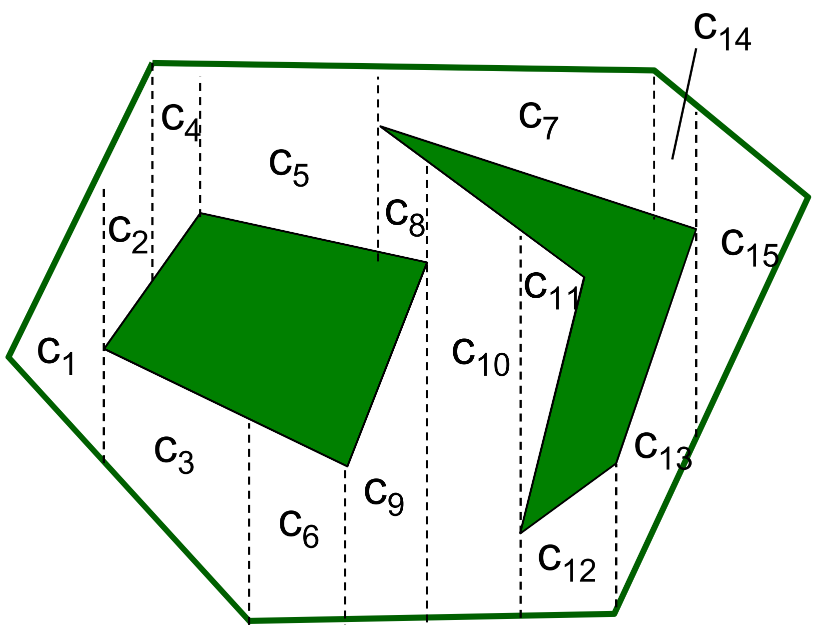
Point robot

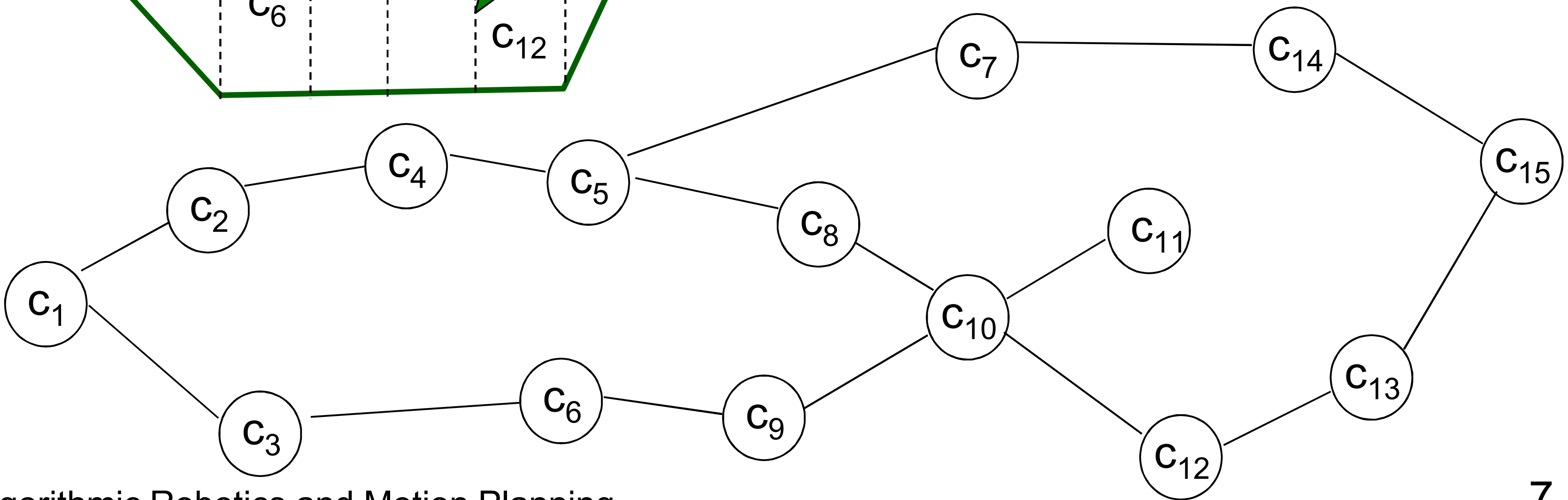


Trapezoidal decomposition



Connectivity graph





What is the number of DoF’s?

- a polygon robot translating in the plane
- a polygon robot translating and rotating
- a spherical robot moving in space
- a spatial robot translating and rotating
- a snake robot in the plane with 3 links

Two major planning frameworks

- cell decomposition
- road map

- motion planning methods differ along additional parameters

Hardness

- the problem is hard when k is part of the input [Reif 79], [Hopcroft et al. 84], …
- [Reif 79]: planning a free path for a robot made of an arbitrary number of polyhedral bodies connected together at some joint vertices, among a finite set of polyhedral obstacles, between any two given configurations, is a PSPACE-hard problem
- translating rectangles, planar linkages

Complete solutions, I

the Piano Movers series [Schwartz-Sharir 83],
cell decomposition: a doubly-exponential solution, $O((nd)^{3^k})$ expected time

assuming the robot complexity is constant,
n is the complexity of the obstacles and
d is the algebraic complexity of the problem

Complete solutions, II

roadmap [Canny 87]:

a singly exponential solution,

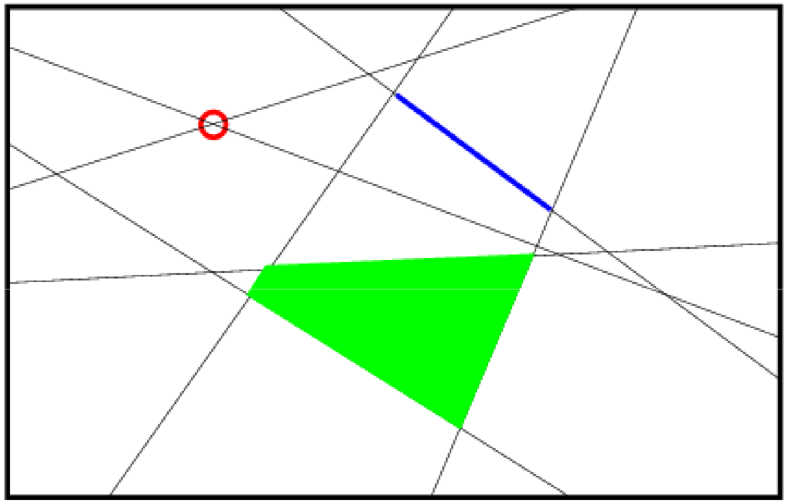
$n^k(\log n)d^{O(k^2)}$ expected time

And now to something completely different

(temporarily different)

What are arrangements?

Example: an arrangement of lines



vertex

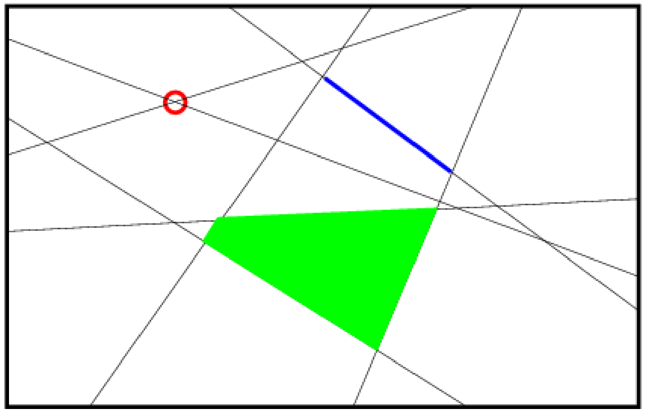
edge

face

What are arrangements, cont'd

- an arrangement of a set S of geometric objects is the subdivision of space where the objects reside induced by S
- possibly non-linear objects (parabolas), bounded objects (segments, circles), higher dimensional (planes, simplices)
- numerous applications in robotics, molecular biology, vision, graphics, CAD/CAM, statistics, GIS
- have been studied for decades, originally mostly combinatorics
  nowadays mainly studied in combinatorial and computational geometry

Arrangements of lines: Combinatorics



the complexity of an arrangement is the overall number of cells of all dimensions comprising the arrangement

for planar arrangements we count: vertices, edges, and faces

the general position assumption: two lines meet in a single point, three lines have no point in common

In an arrangements of n lines

number of vertices: $n(n-1)/2$

number of edges: n^2

number of faces:

using Euler's formula $|V|-|E|+|F|=2$
we get $n^2+n^2/2+1$

Basic theorem of arrangement complexity

the maximum combinatorial complexity of an arrangement of n well-behaved curves in the plane is $O(n^2)$; there are such arrangements whose complexity is $\Omega(n^2)$.

more generally

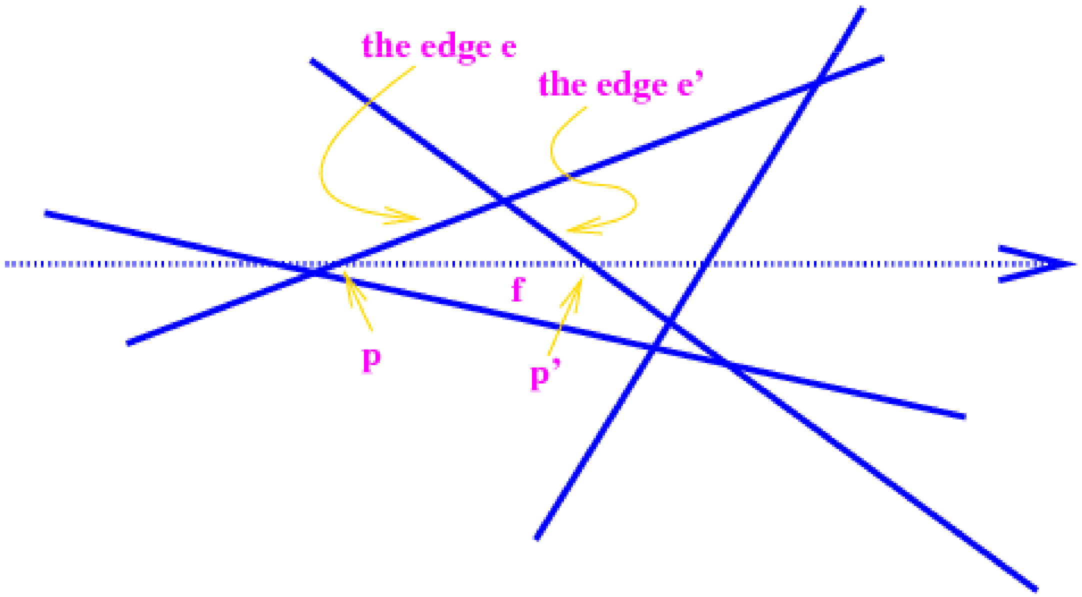
the maximum combinatorial complexity of an arrangement of n well-behaved (hyper)surfaces in $\mathbb{R}^d$ is $O(n^d)$; there are such arrangements whose complexity is $\Omega(n^d)$

Constructing the arrangement

- representation (data structure)
  - DCEL - the doubly connected edge list
- algorithm
  - incremental
- computational model
  - the real RAM
- the general position assumption

Incremental construction

- computing a bounding box
- inserting the ith line



How much time does it take?

the steps:

- computing the bounding box

naively $O(n^2)$; can be done in $O(n \log n)$, exercise

- finding where to insert line i

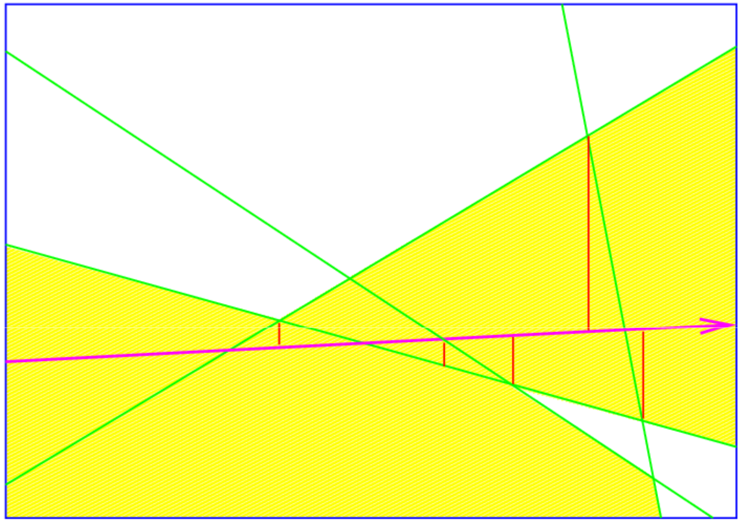
simple, $O(i)$

- inserting line i

O(zone complexity)

The zone of a curve

the zone of a curve γ in an arrangement $\mathcal{A}$ is the collection of faces of $\mathcal{A}$ intersected by γ



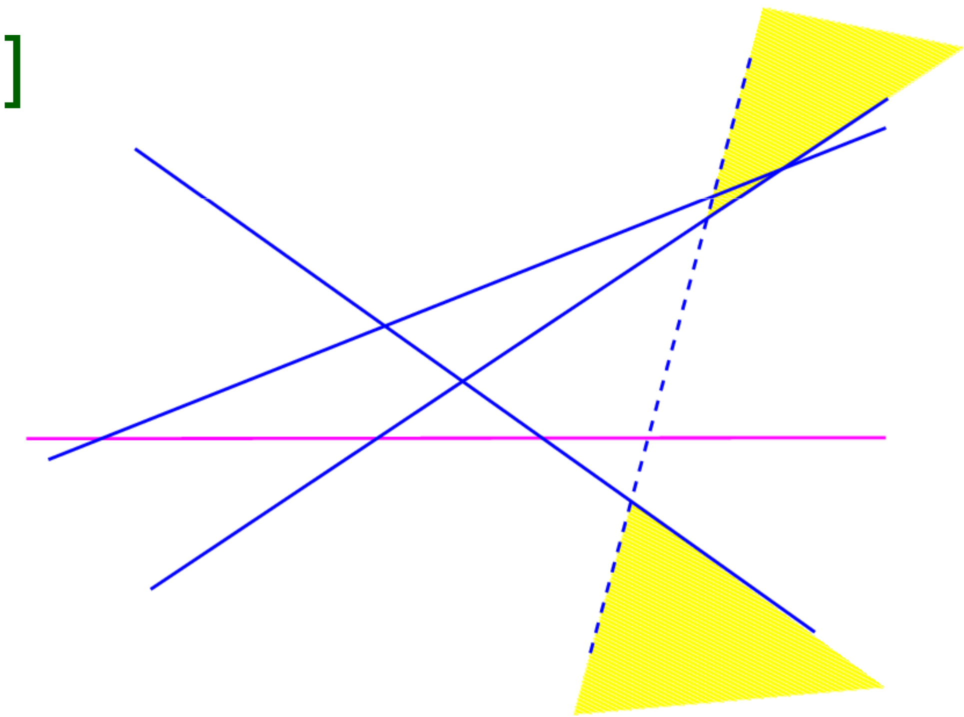
the complexity of the zone is the overall complexity of cells of various dimensions in the closure of the zone

we need: the complexity of the zone of a line in an arrangement of i lines

Zone theorem

theorem: the complexity of the zone of a line in an arrangement of i lines is $O(i)$

proof [omitted]



Overall running time of the algorithm

- computing the bounding box

naively $O(n^2)$; can be done in $O(n \log n)$

- finding where to insert line i

simple, $O(i)$

- inserting line i

O(zone complexity) = $O(i)$

overall $O(n^2)$ time

Basics of arrangements, summary

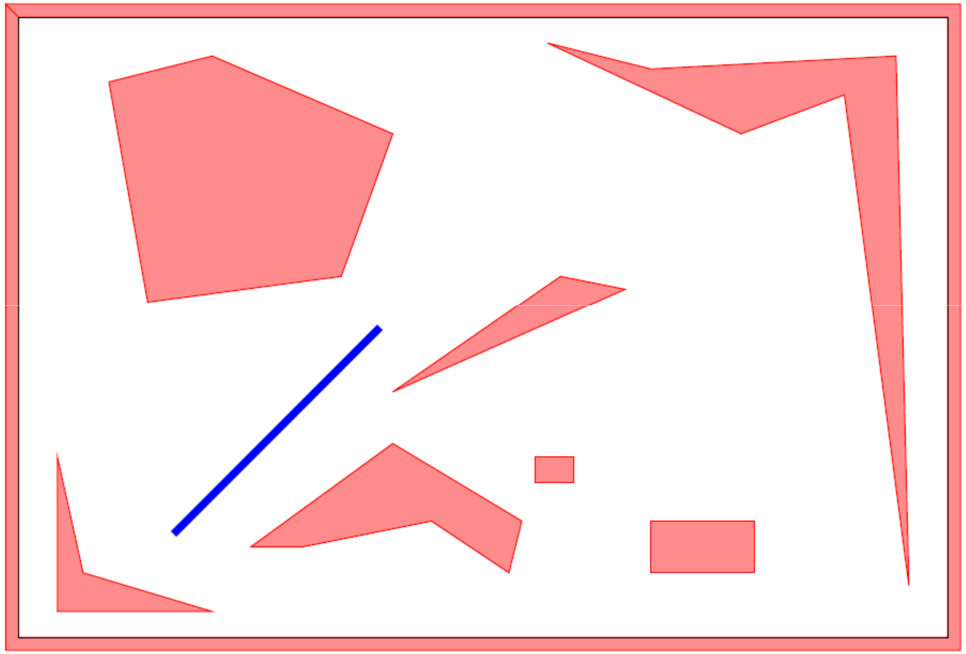
- the combinatorial complexity of an arrangement
- incremental construction
- the zone of an(other) object in an arrangement
- the basic theorem of arrangement complexity
- the real RAM model, the general position assumption

Configuration spaces

- arrangements $\mathcal{A}(\mathcal{S})$ are used for exact discretization of continuous problems
- a point p in configuration space $\mathcal{C}$ has a property $\Pi(p)$
- if a neighborhood U of p is not intersected by an object in $\mathcal{S}$, the same property $\Pi(q)$ holds for every point $q \in U$
  (the same holds when we restrict the configuration space to an object in $\mathcal{S}$)
- the objects in $\mathcal{S}$ are critical
- the property is invariant in each cell of the arrangement

Configuration space for translational motion planning

the rod is translating in the room



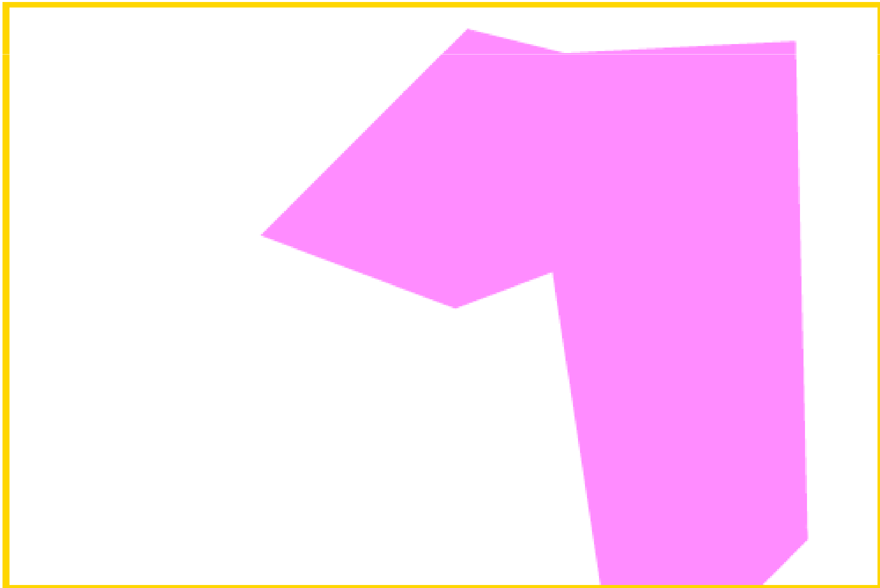
- the reference point: the lower end-point of the rod
- the configuration space is 2 dimensional

Configuration space obstacles

the robot has shrunk to a point

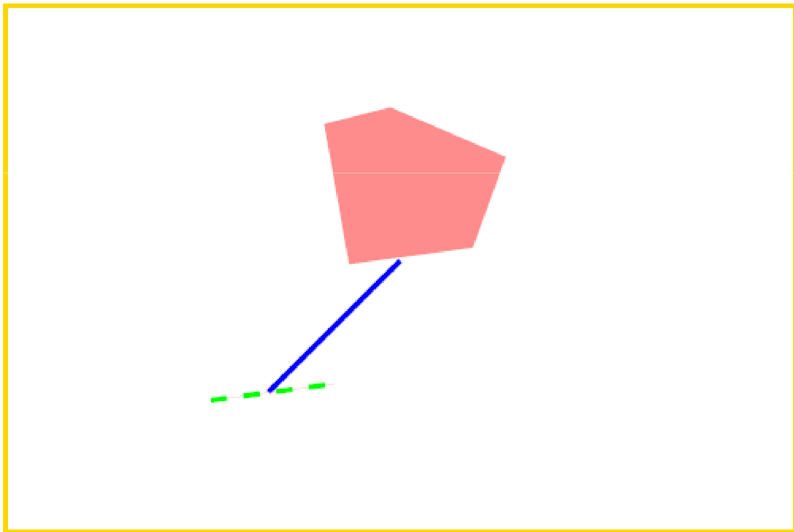
⇒

the obstacles are accordingly expanded



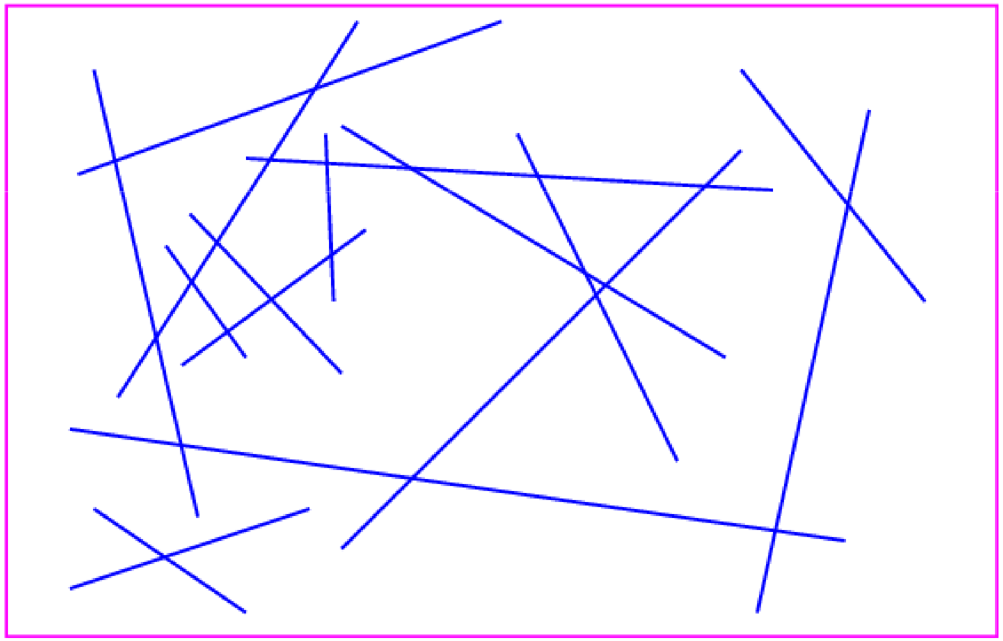
Critical curves in configuration space

the locus of semi-free placements



Making the connection:

The arrangement of critical curves in configuration space

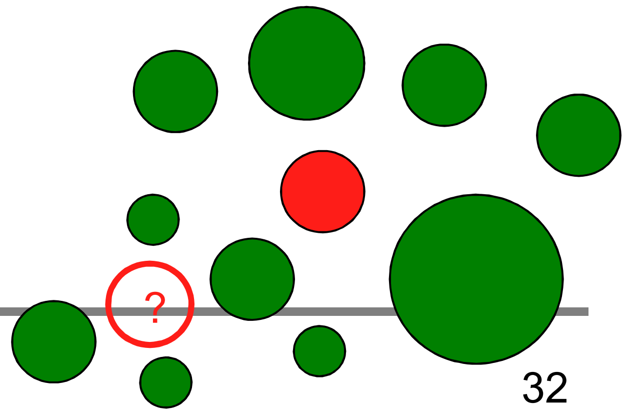


Solving a motion-planning problem
a general framework

- what are the critical curves
- how complex is the arrangement of the critical curves
- constructing the arrangement and filtering out the forbidden cells
- what is the complexity of the free space
- can we compute the free space efficiently
- do we need to compute the entire free space?

Example: a disc moving among discs

- the critical curves are circles
- how complex is the arrangement of the circles?
- what is the complexity of the free space?
- can we compute the free space efficiently?
- do we need to compute the entire free space? does it matter?



Example: an L-shaped robot moving among points

- what are the critical curves?
- how complex is the arrangement of the critical curves?
- what is the complexity of the free space?
- how to compute the free space efficiently?
- do we need to compute the entire free space? does it matter?

THE END