Robot Autonomy

Lecture 5:
Planning Problems and Configuration Spaces

Oliver Kroemer

An Incomplete Taxonomy of Motion Planning

Plan Types

- Discrete
  - Discrete sets of states and actions
  - STanford Research Institute Problem Solver
- Continuous
  - Piano movers' problem
  - Grasp planning
- Hybrid
  - Attributes of both discrete and continuous planning
  - Which skills to perform and how to perform them

Environment Types

- Immovable
  - Fixed obstacles

- Movable
  - Objects can be moved through interactions

- Moving
  - Objects move on their own

Interaction Types

- Non-contact interactions
  - Avoid contact with objects, obstacles, environment
- Contact interactions
  - Grasping, climbing, pushing, hanging, etc.

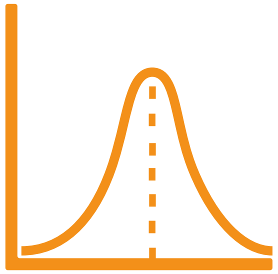
State Uncertainty Types

- None

s

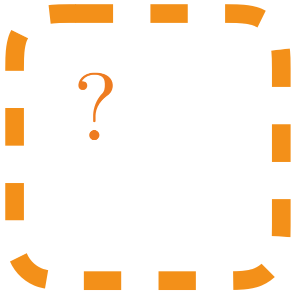
- Probabilistic

$p(s)$



- Bounded

$s \in S$



- Unknown

$s = ???$

???

Robot Motion Types

- Kinematic
  - Find a path

$$s = \left[\; q \; \right] \qquad \tau : [0,1] \rightarrow \mathbb{S} \qquad \begin{aligned} \tau(1) &= q_g \\ \tau(0) &= q_i \end{aligned}$$

- Dynamic
  - Find a trajectory

$$s = \begin{bmatrix} q \\ \dot{q} \end{bmatrix} \qquad \tau : [0, \underbrace{T}_{\text{Time}}] \rightarrow \mathbb{S} \qquad \begin{aligned} \tau(0) &= \begin{bmatrix} q_i \\ \dot{q}_i \end{bmatrix} \\ \tau(T) &= \begin{bmatrix} q_g \\ \dot{q}_g \end{bmatrix} \end{aligned}$$

Constraint Types

- Holonomic
  - Can be expressed as a function of the system's configuration

$$f(q,t) = 0$$

- Non-Holonomic
  - Cannot be defined/reduced to the above holonomic form
  - Constraints on moving through the configuration space
    - Manipulating different objects
    - Driving a car

$$f(q,\dot{q},t) = 0$$

Configuration Spaces

Based on slides of Katharina Muelling

Zoo of Robot


NASA


Amazon Robotics


NASA JPL

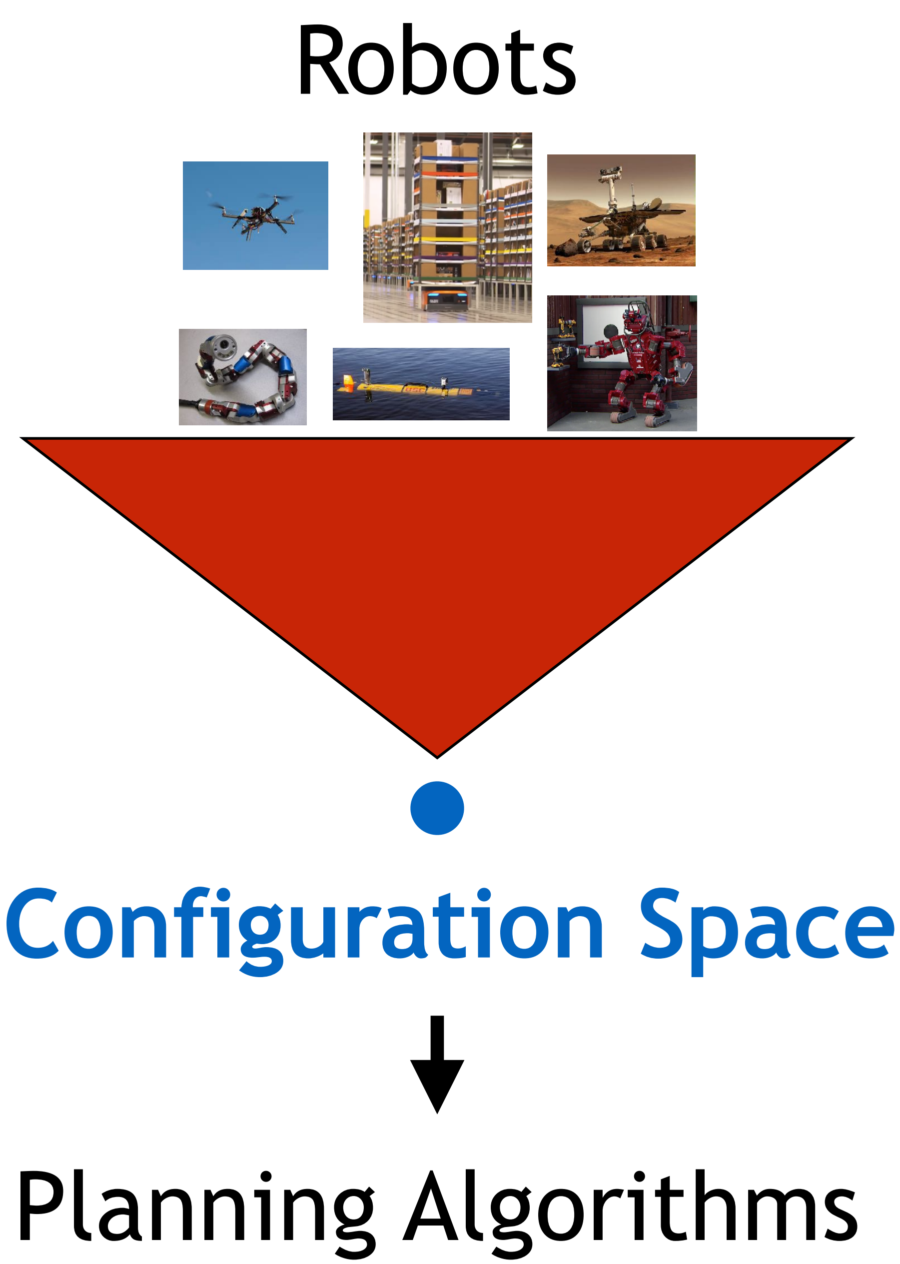

CMU BioRobot


USC RESL


Darpa Challenge

Unified Configuration Representation

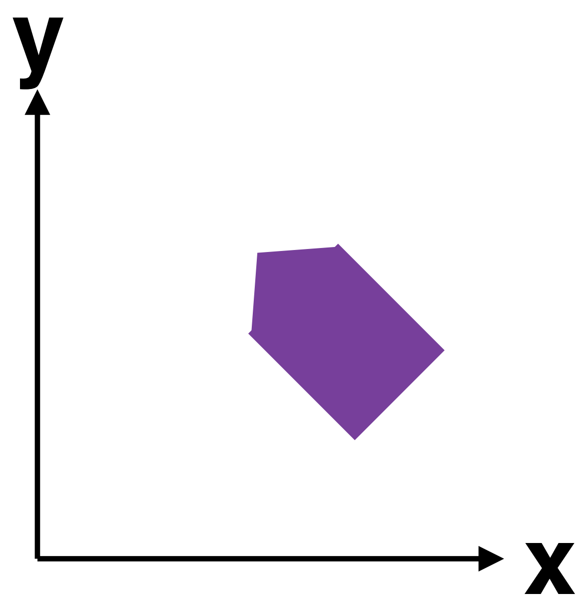
- Do not want to create robot-specific algorithms
- Define a space in which all robots are defined as a point



Workspace

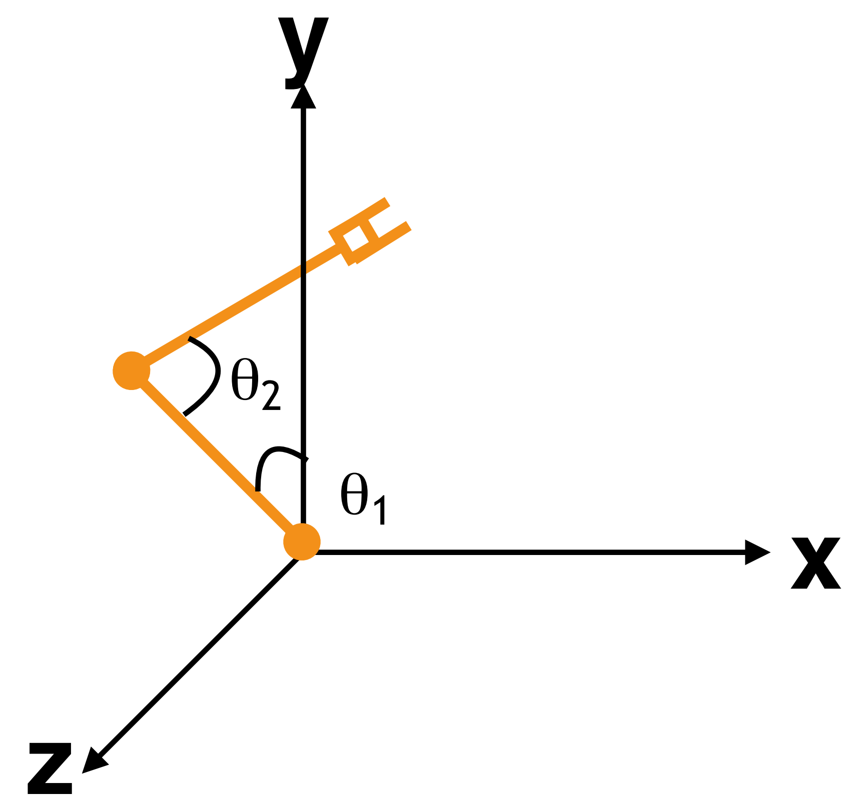
The workspace is defined as the world in which the robot exists and occupies space

2D workspace



3-parameter specification:
(x, y, θ)

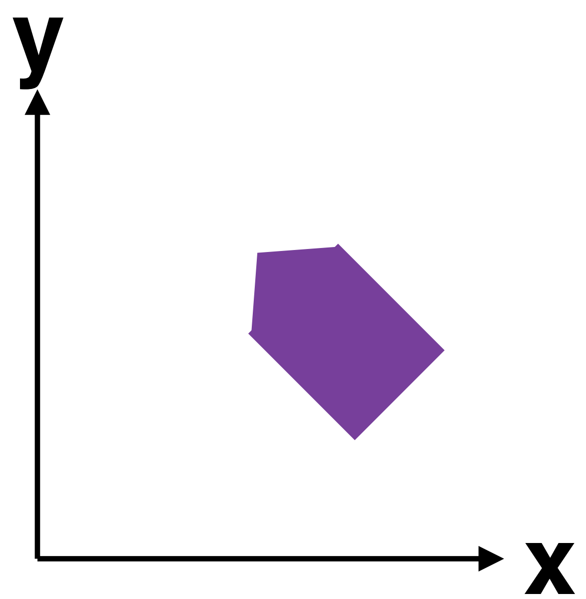
3D workspace



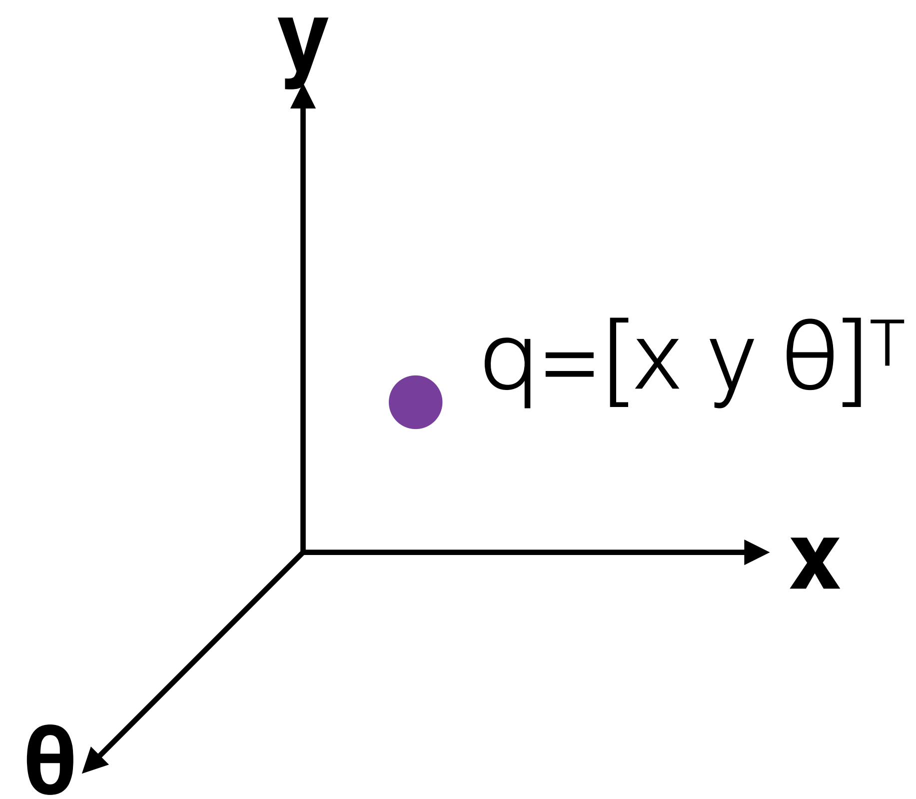
6-parameter specification:
$(x, y, z, \alpha, \beta, \gamma)$

Configuration Space

A configuration q is the sufficient and complete specification of the position of every point on the physical system



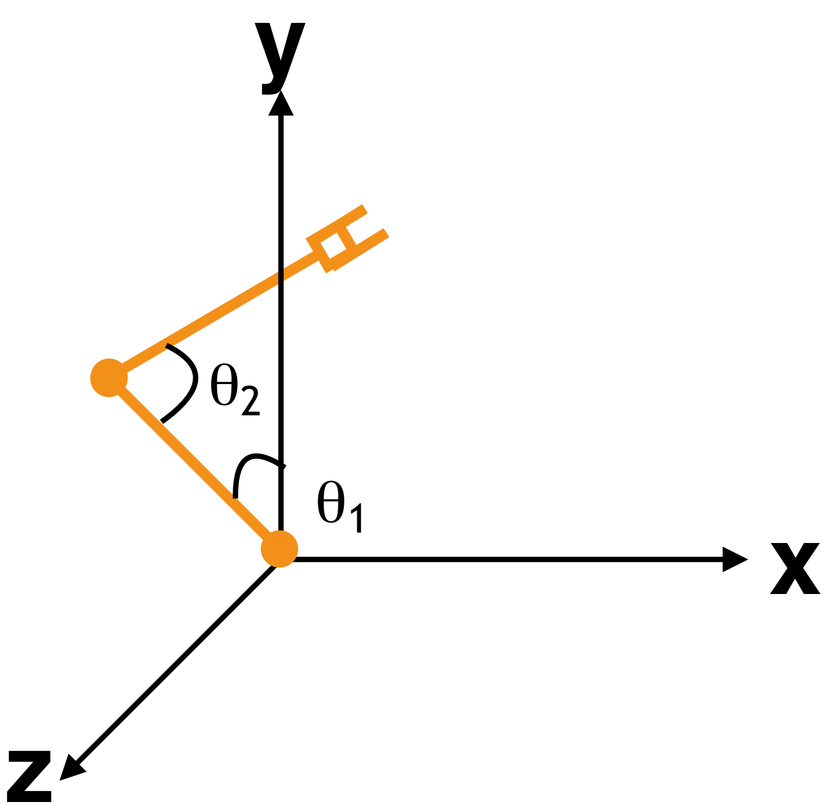
Workspace



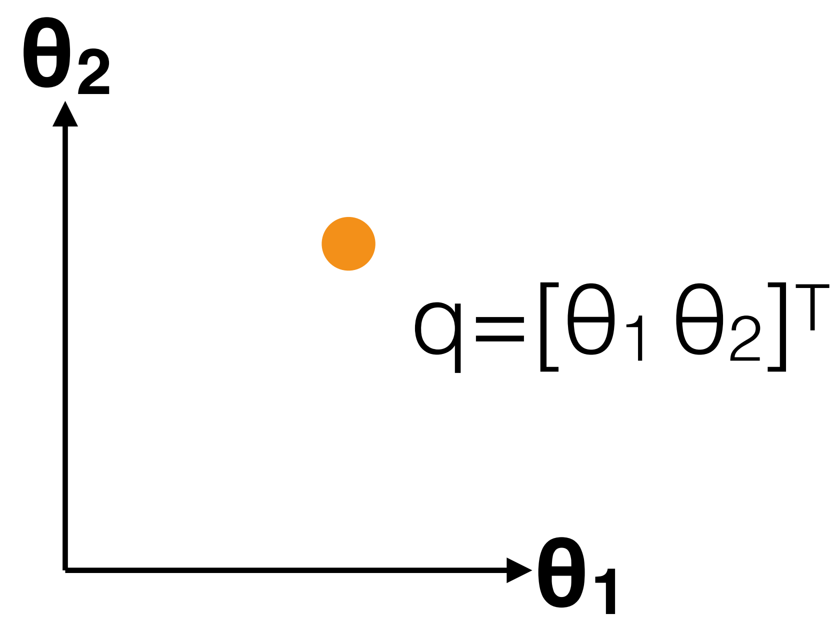
Configuration Space (C-Space)

Configuration Space

A configuration q is the sufficient and complete specification of the position of every point on the physical system



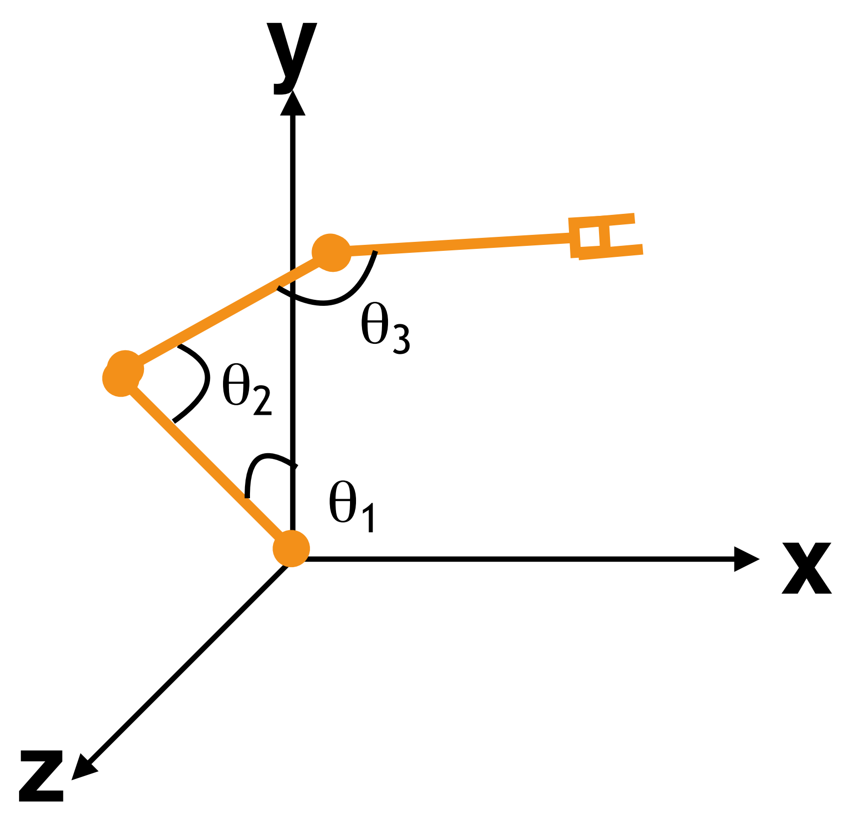
Workspace



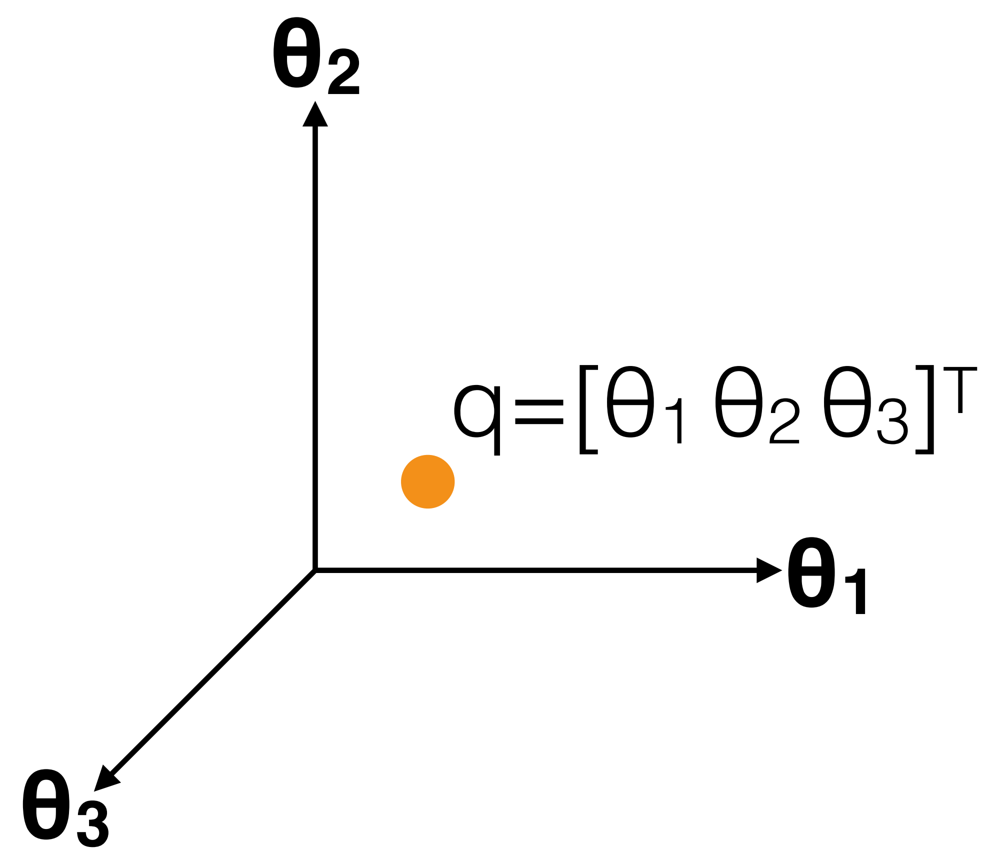
Configuration Space
(C-Space)

Configuration Space

A configuration q is the sufficient and complete specification of the position of every point on the physical system



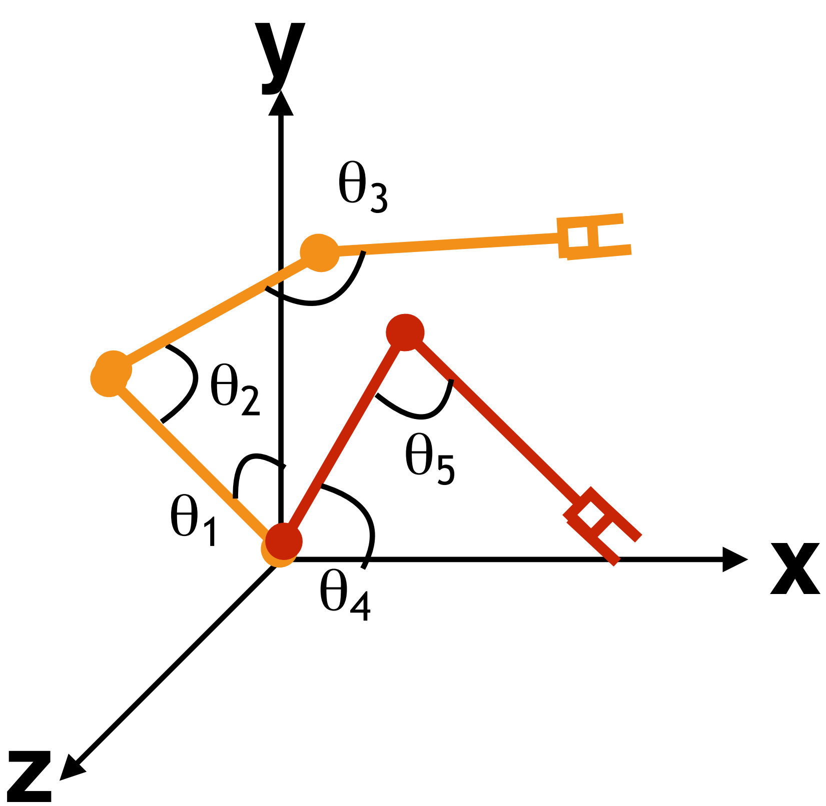
Workspace



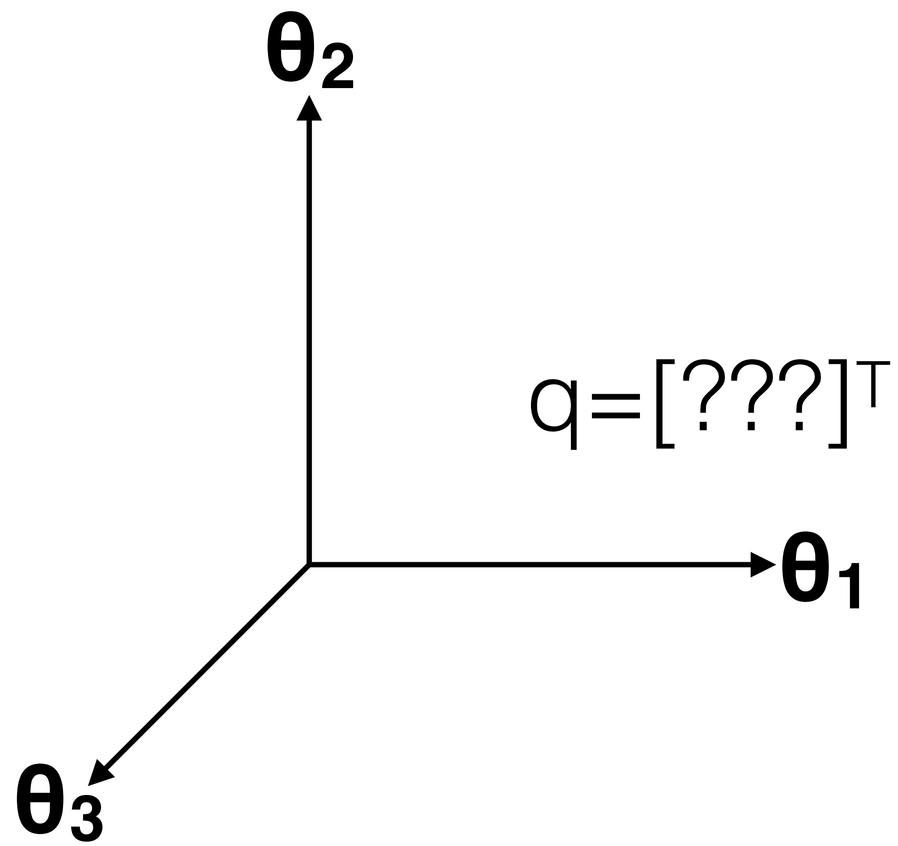
Configuration Space
(C-Space)

Configuration Space

A configuration q is the sufficient and complete specification of the position of every point on the physical system



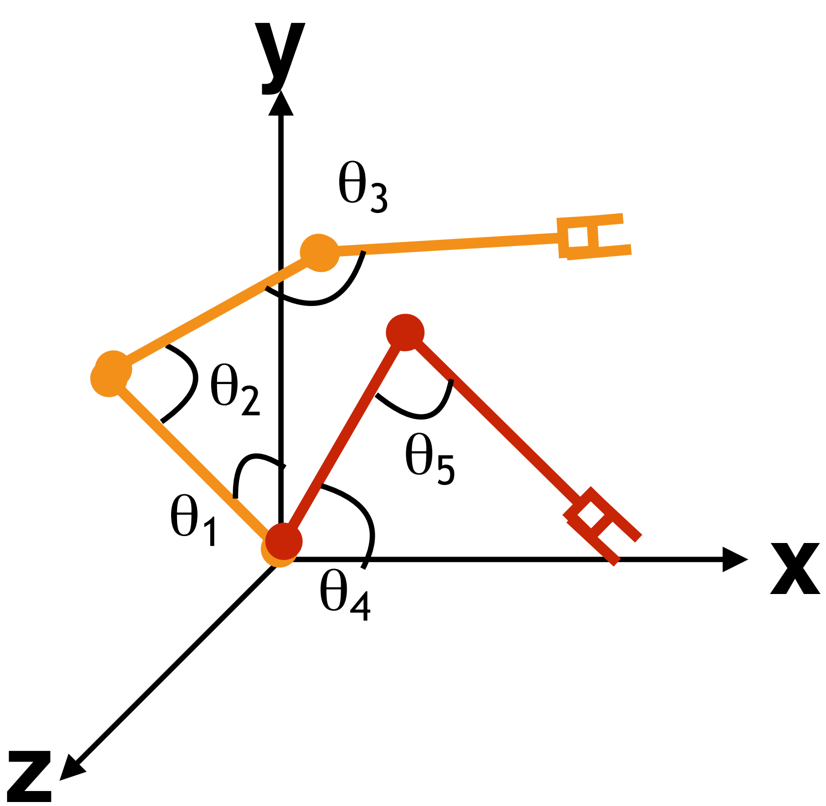
Workspace



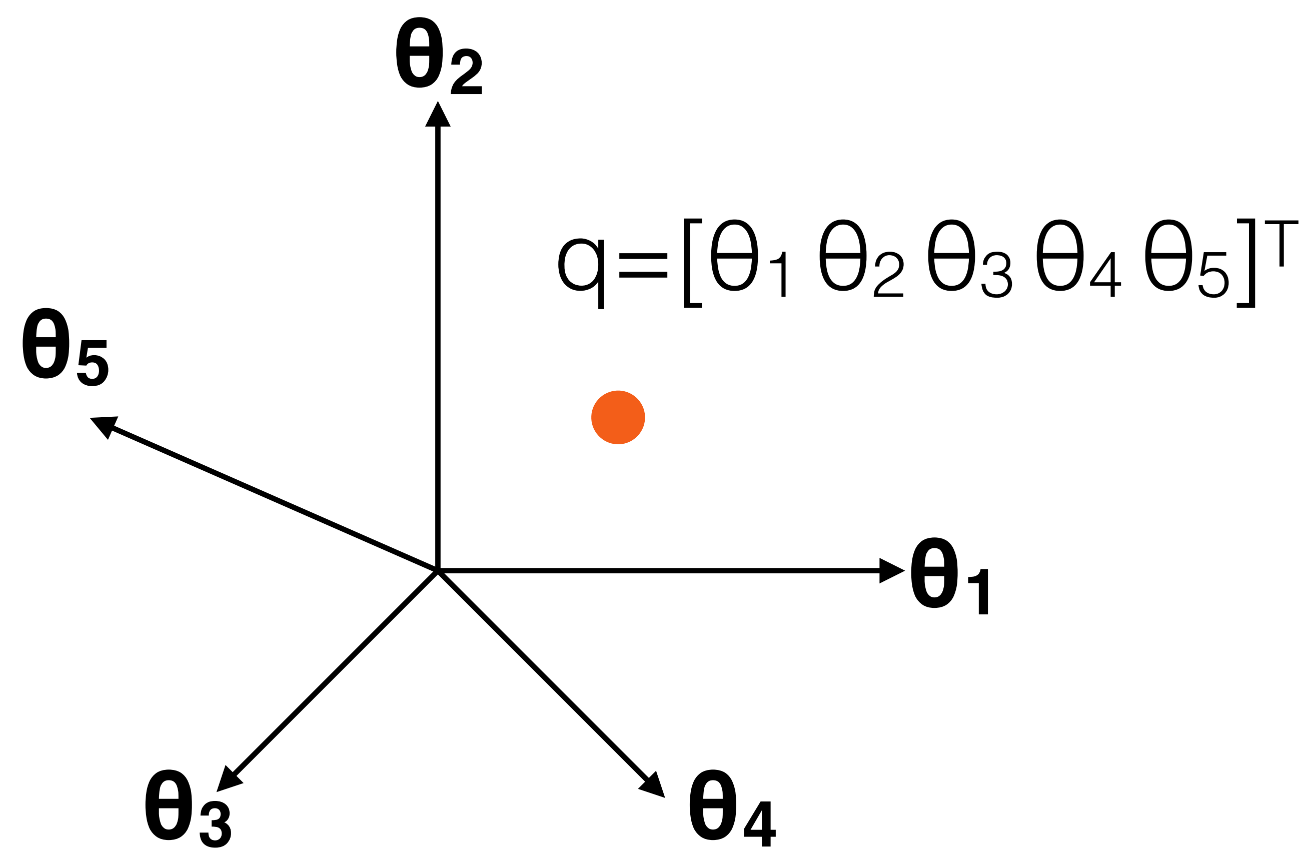
Configuration Space
(C-Space)

Configuration Space

A configuration q is the sufficient and complete specification of the position of every point on the physical system



Workspace



Configuration Space
(C-Space)

Configuration Space

A configuration q is the sufficient and complete specification of the position of every point on the physical system

The configuration space C is the set of all possible configurations such that

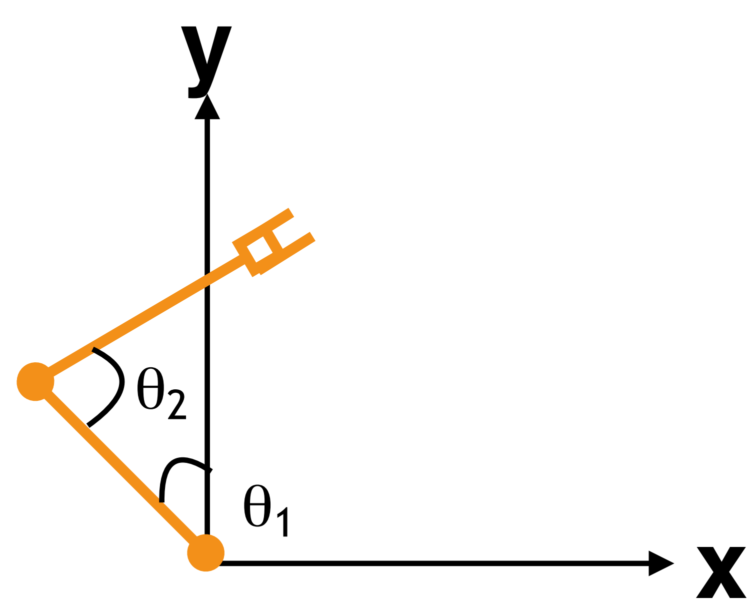
$$q \in C$$

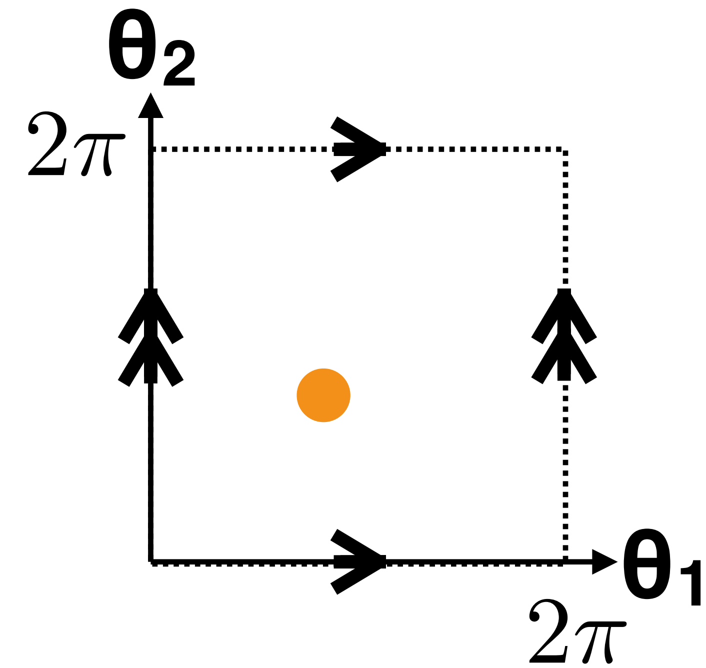
The mapping from c-space to the world space is given by

$$A(q) : C \to W$$

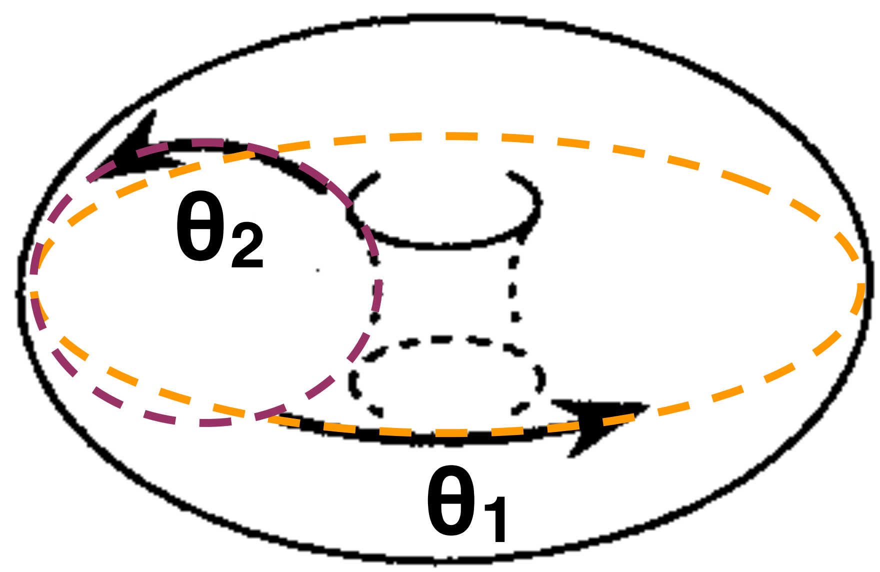
C-Space Topologies

- Configuration spaces do not need to be Cartesian





Torus Topology



$$C = S^1 \times S^1$$

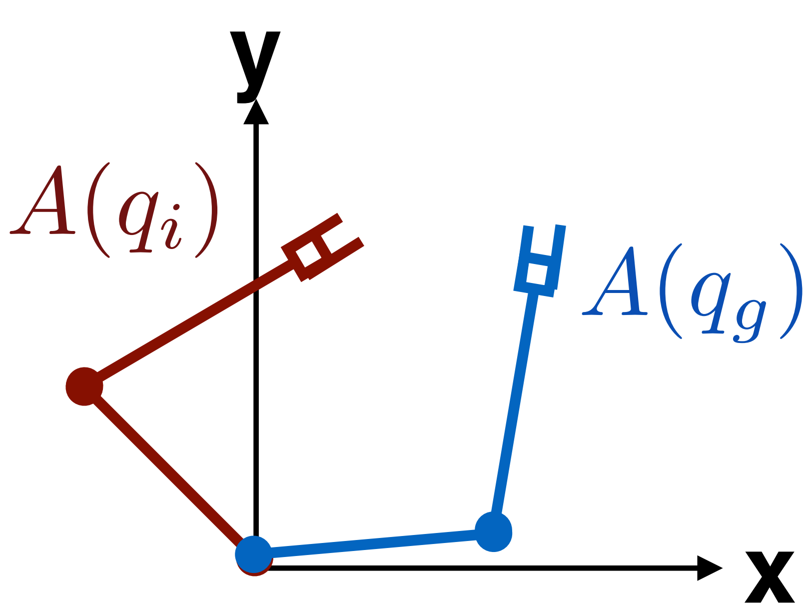
- Shortest distance to a point may involve wrap around!

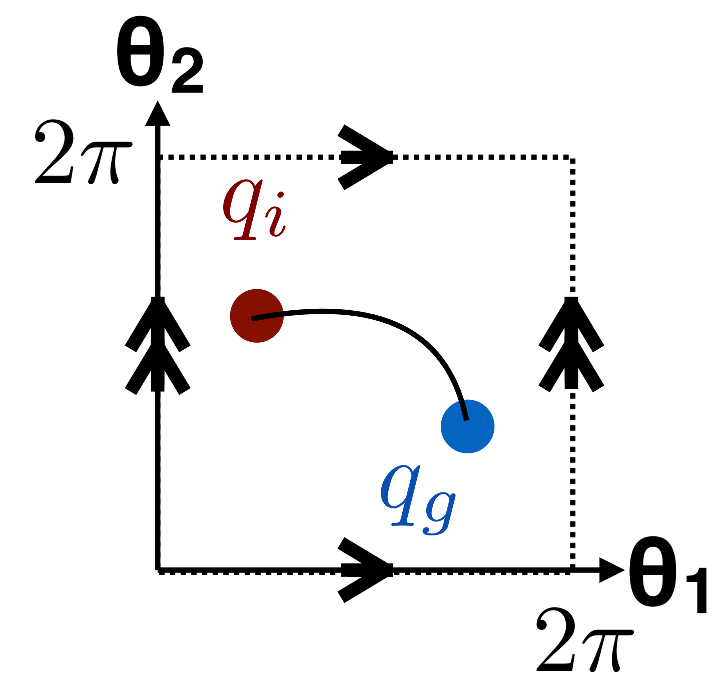
Motion planning problem:

Find a collision-free path in the c-space C

from the initial configuration $q_i \in C$

to the goal configuration $q_g \in C$





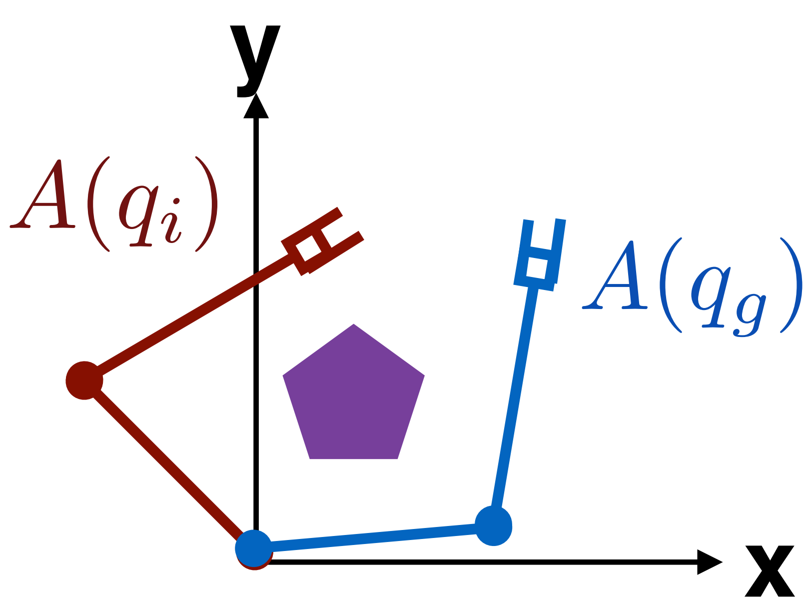
Motion Planning

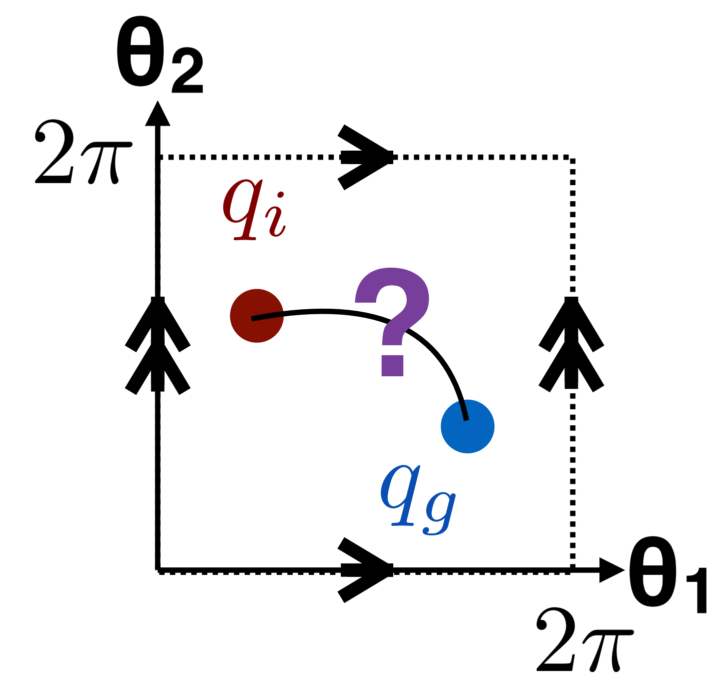
<u>Motion planning problem:</u>

Find a collision-free path in the c-space C

from the initial configuration $q_i \in C$

to the goal configuration $q_g \in C$

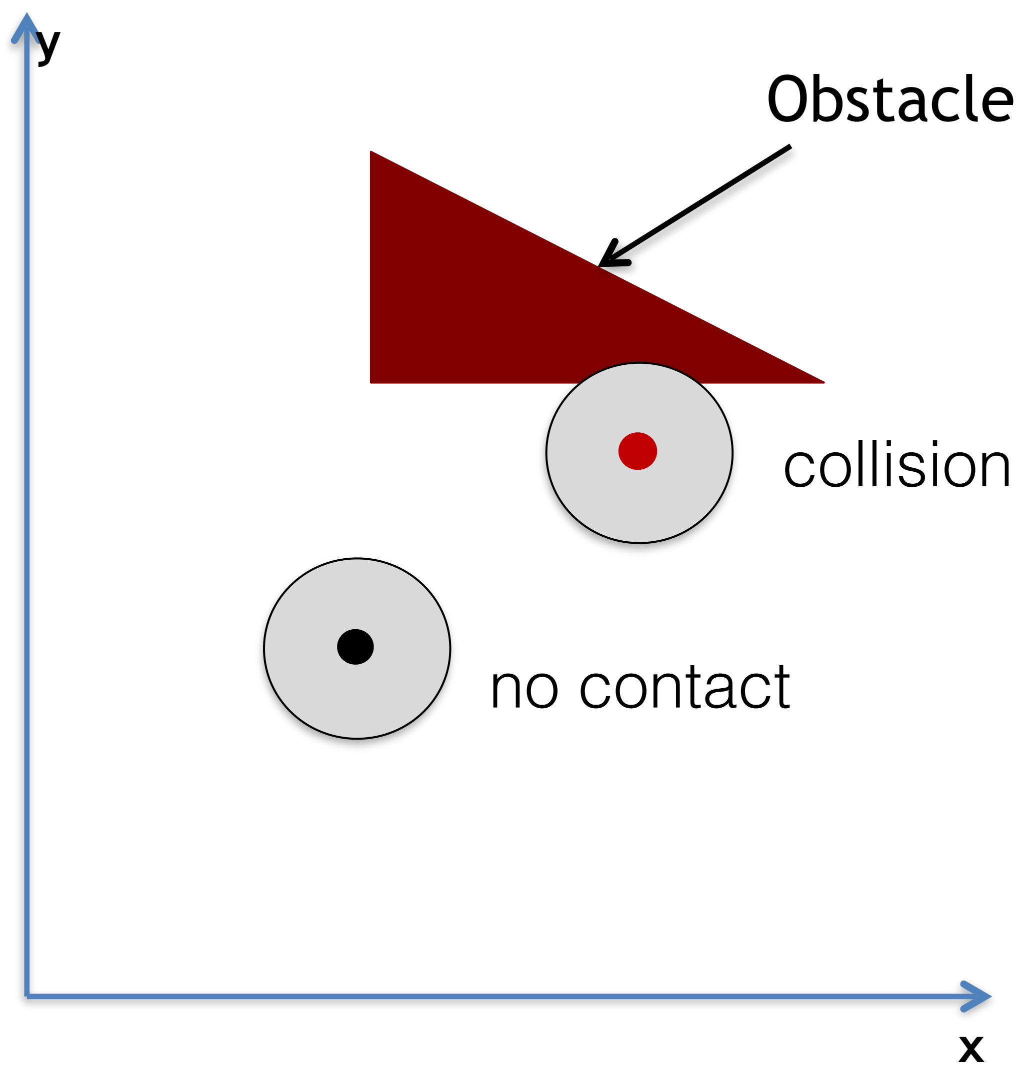




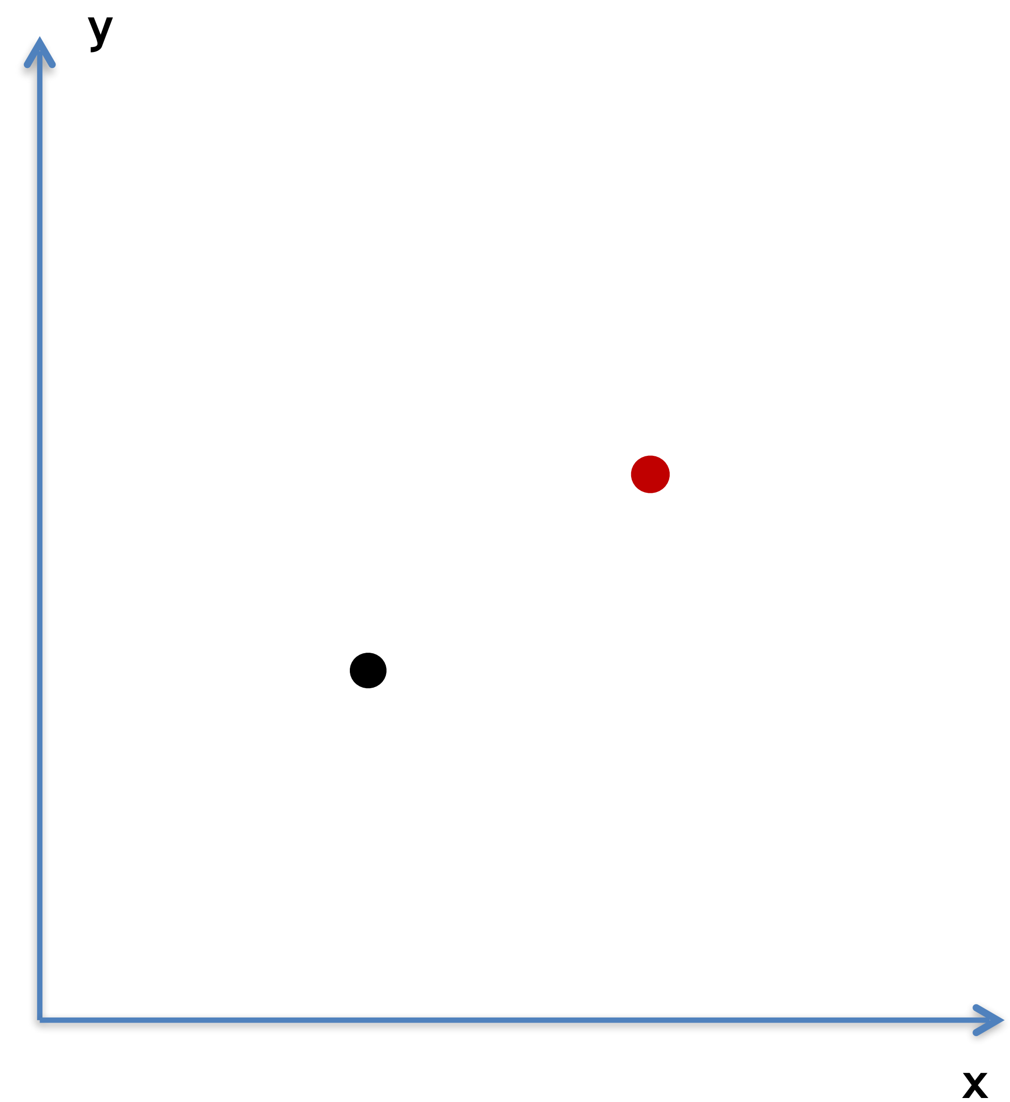
The robot needs to avoid the obstacle region $O \subset W$

What do the obstacles look like in the c-space?

Obstacles

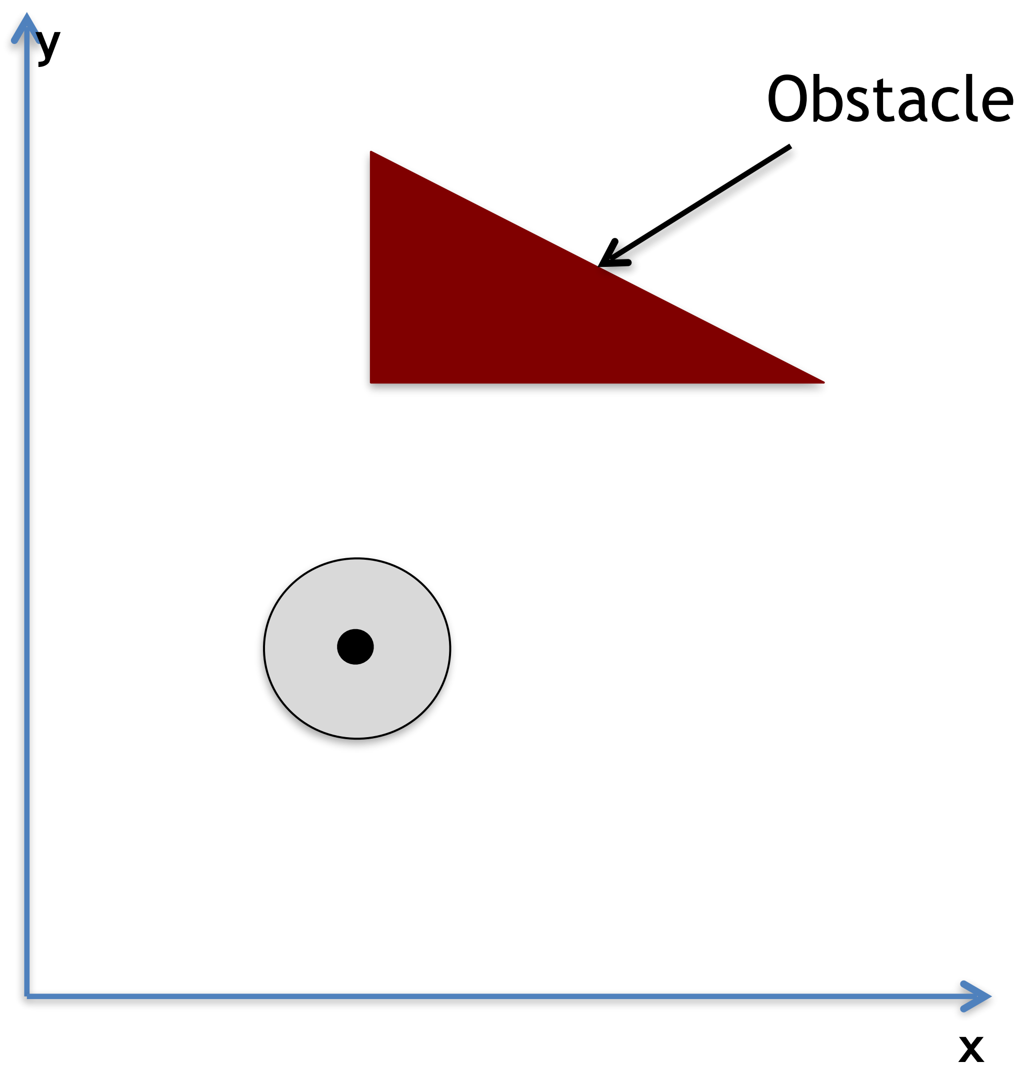


World

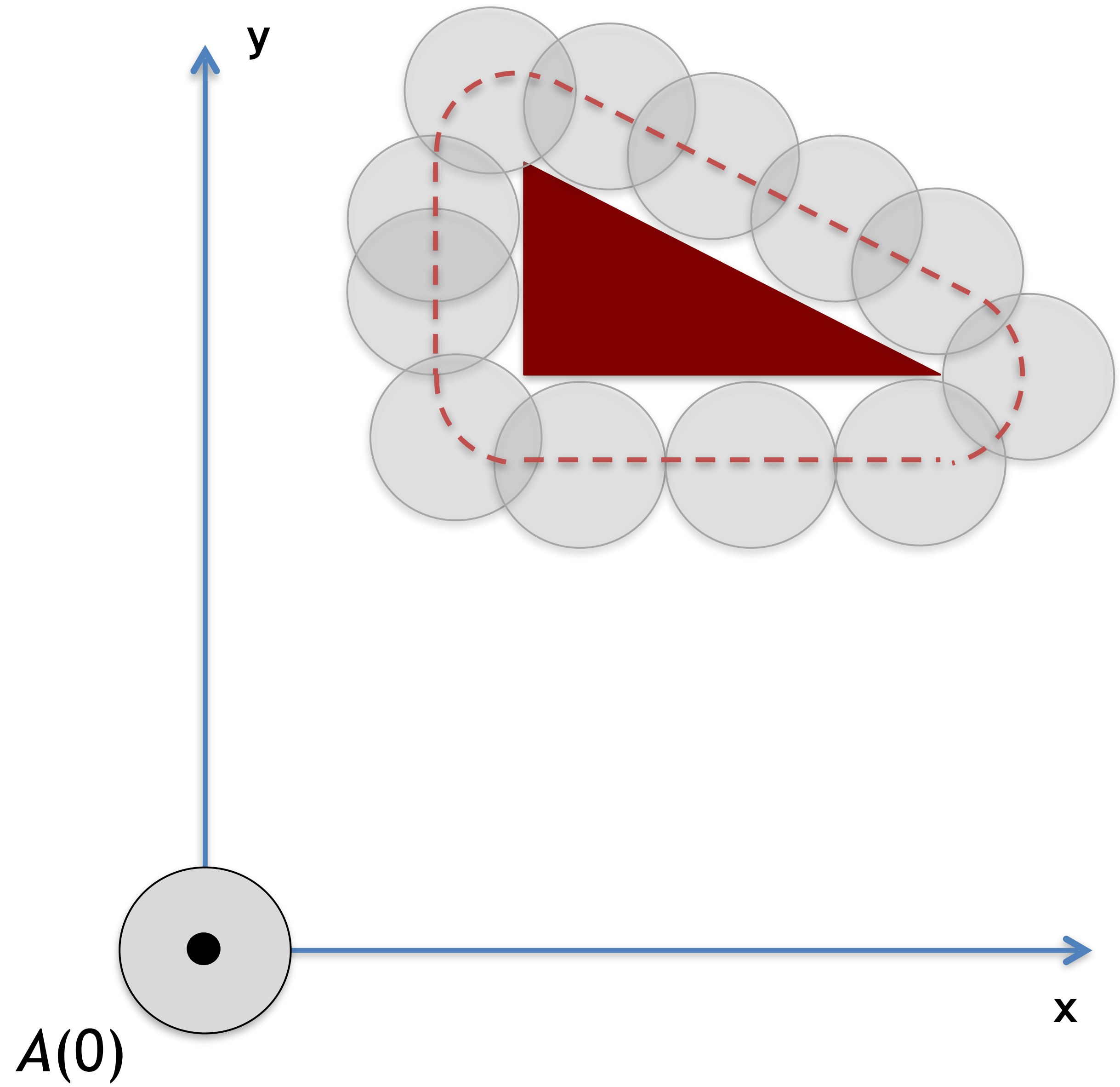


Configuration Space

Obstacles

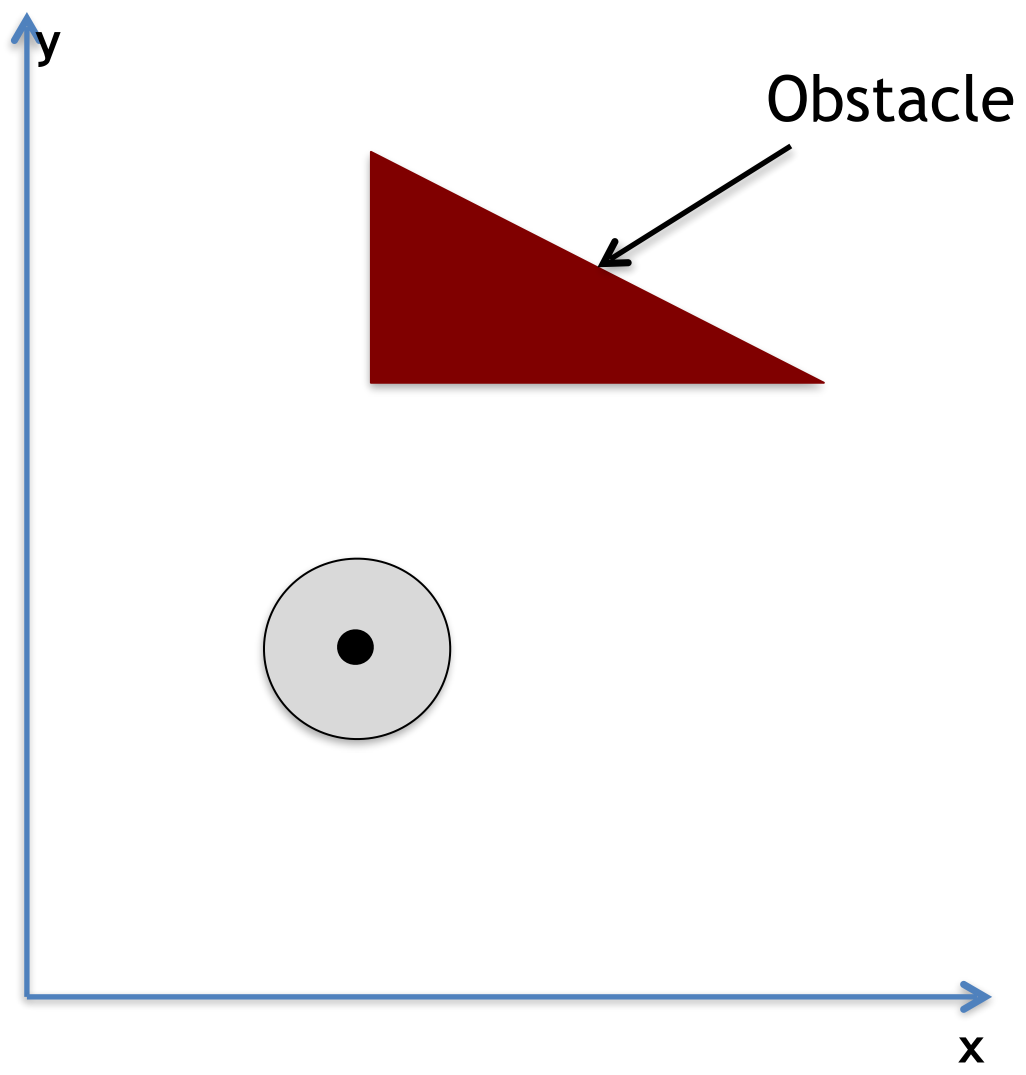


World

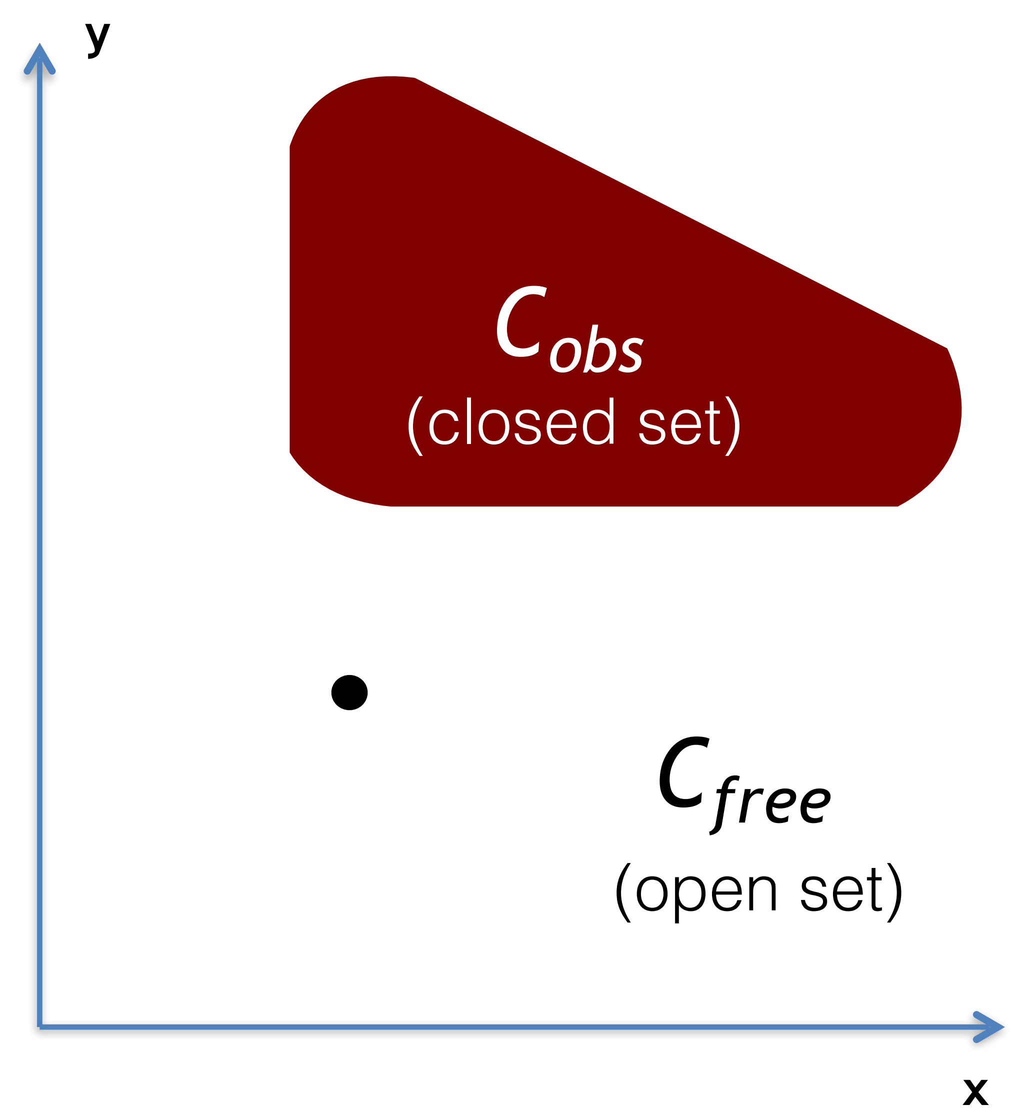


Configuration Space

Obstacles



World



Configuration Space

Minkowski Sum and Difference

We define the obstacle in the configuration space with the Minkowski difference if the robot is rigid and $C = \mathbb{R}^n, n = 1, 2, 3$

Minkowski sum:

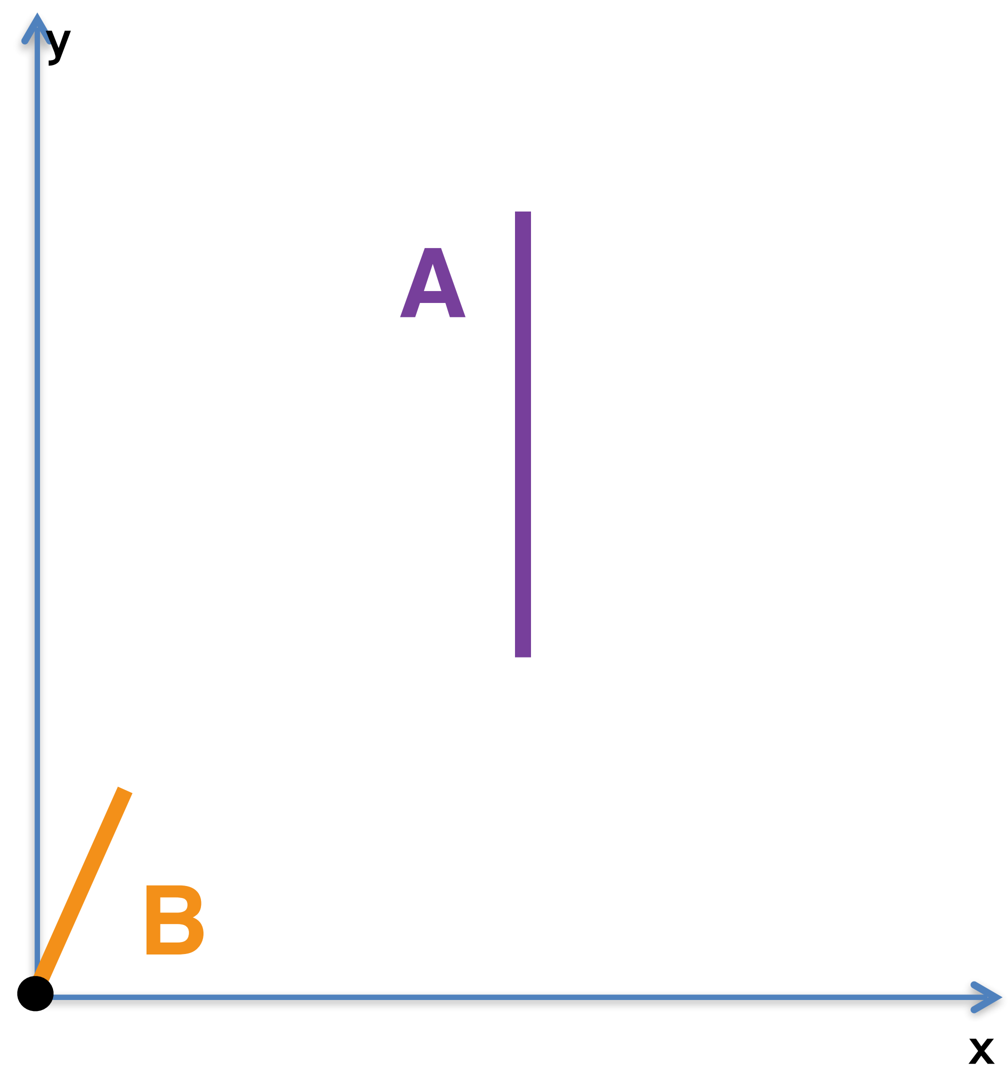
$$A \oplus B = \{a + b \mid a \in A, b \in B\}$$

Minkowski difference:

$$A \ominus B = \{a - b \mid a \in A, b \in B\}$$

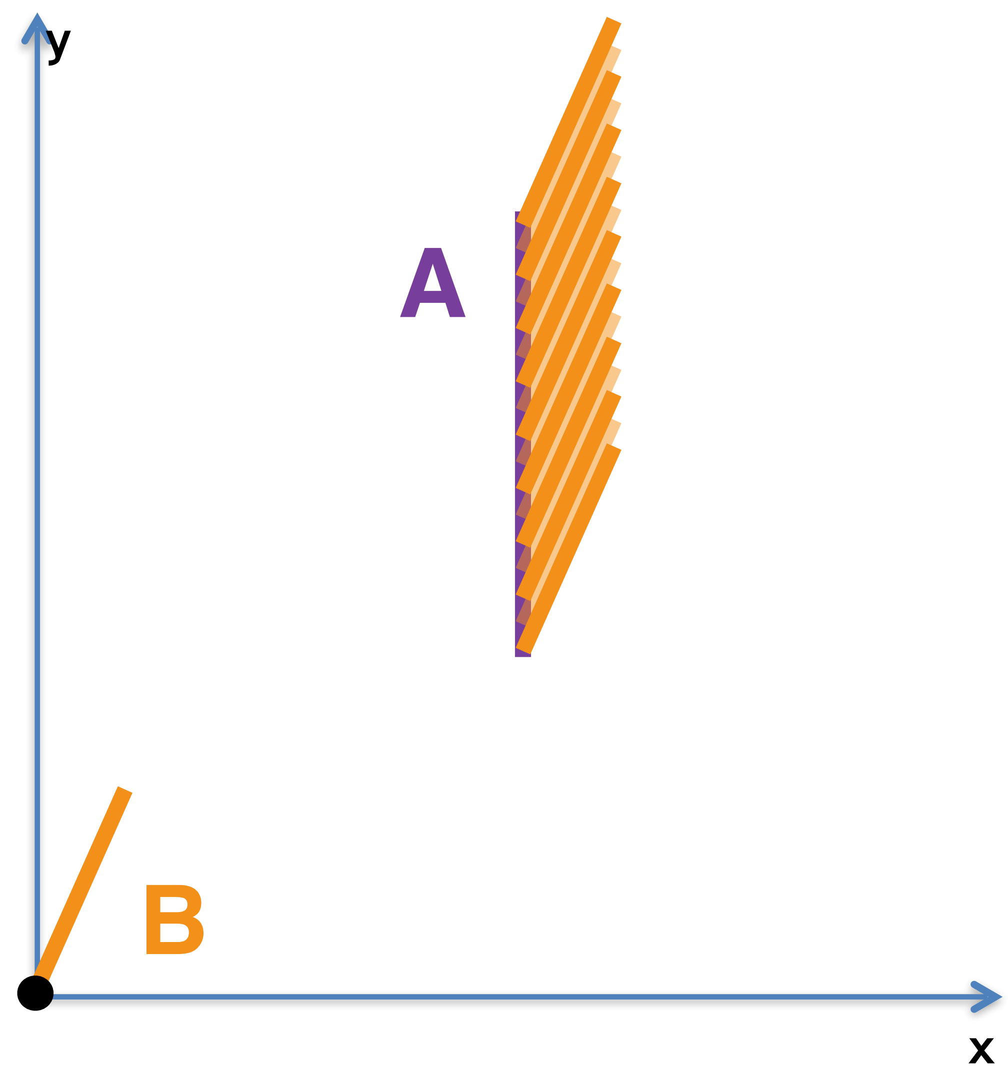
Minkowski Sum

$$A \oplus B = \{a + b \mid a \in A, b \in B\}$$



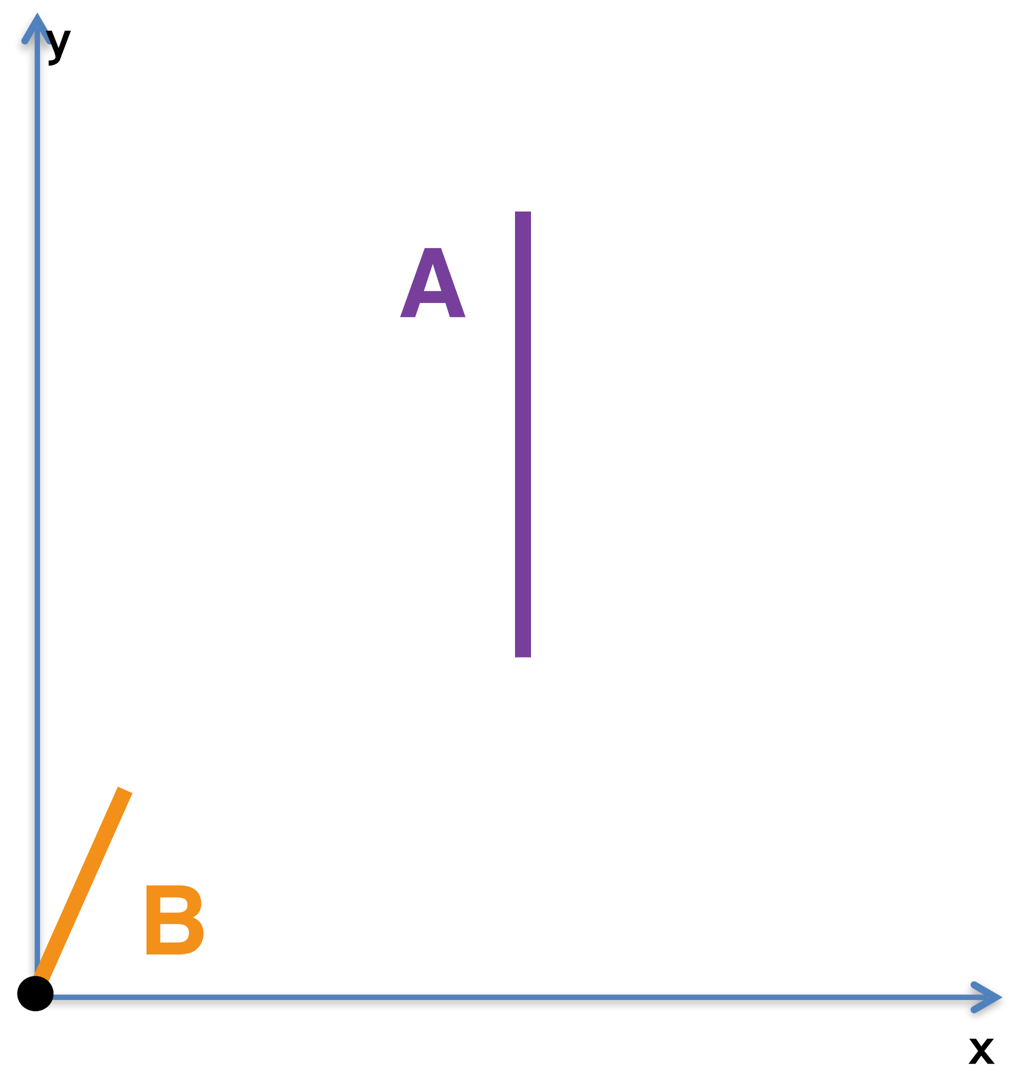
Minkowski Sum

$$A \oplus B = \{a + b \mid a \in A, b \in B\}$$



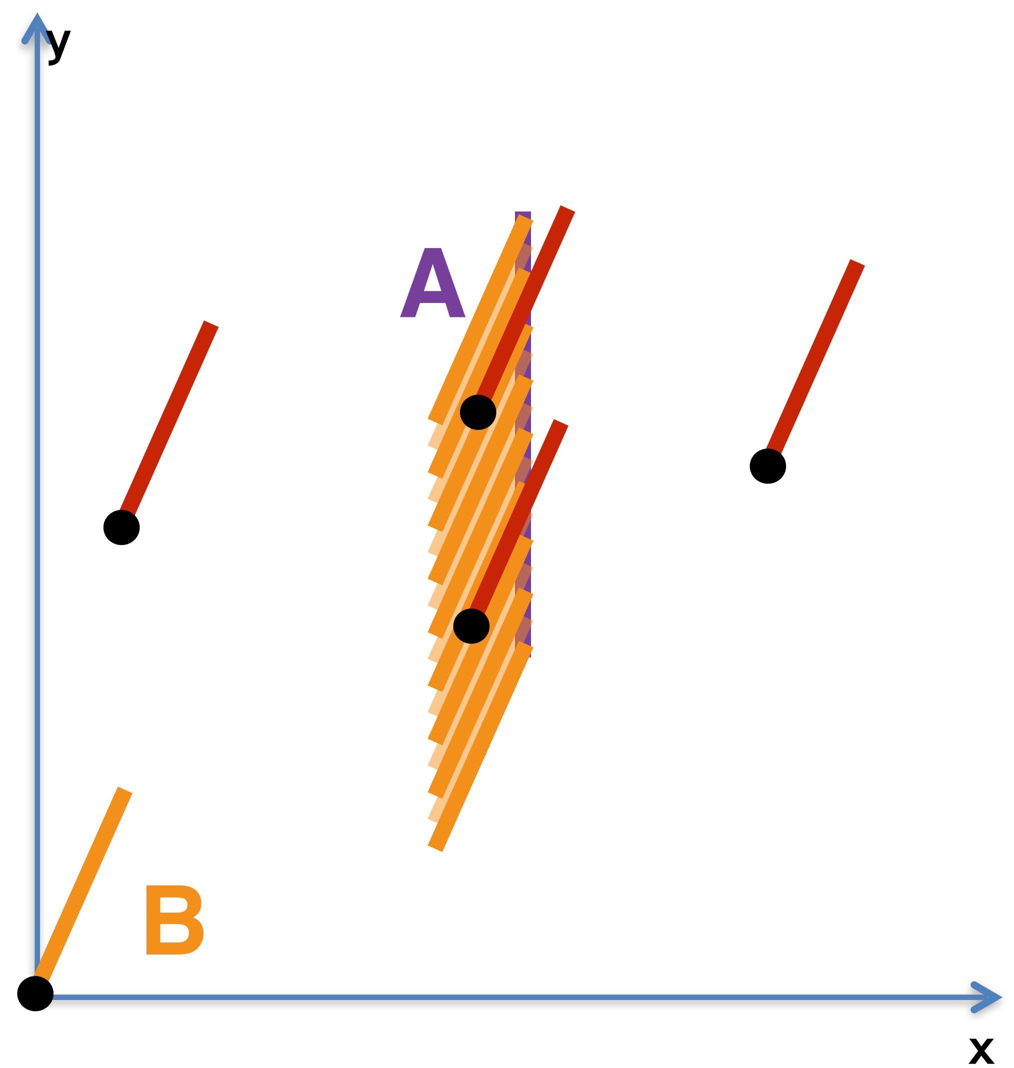
Minkowski Difference

$$A \ominus B = \{a - b \mid a \in A, b \in B\}$$



Minkowski Difference

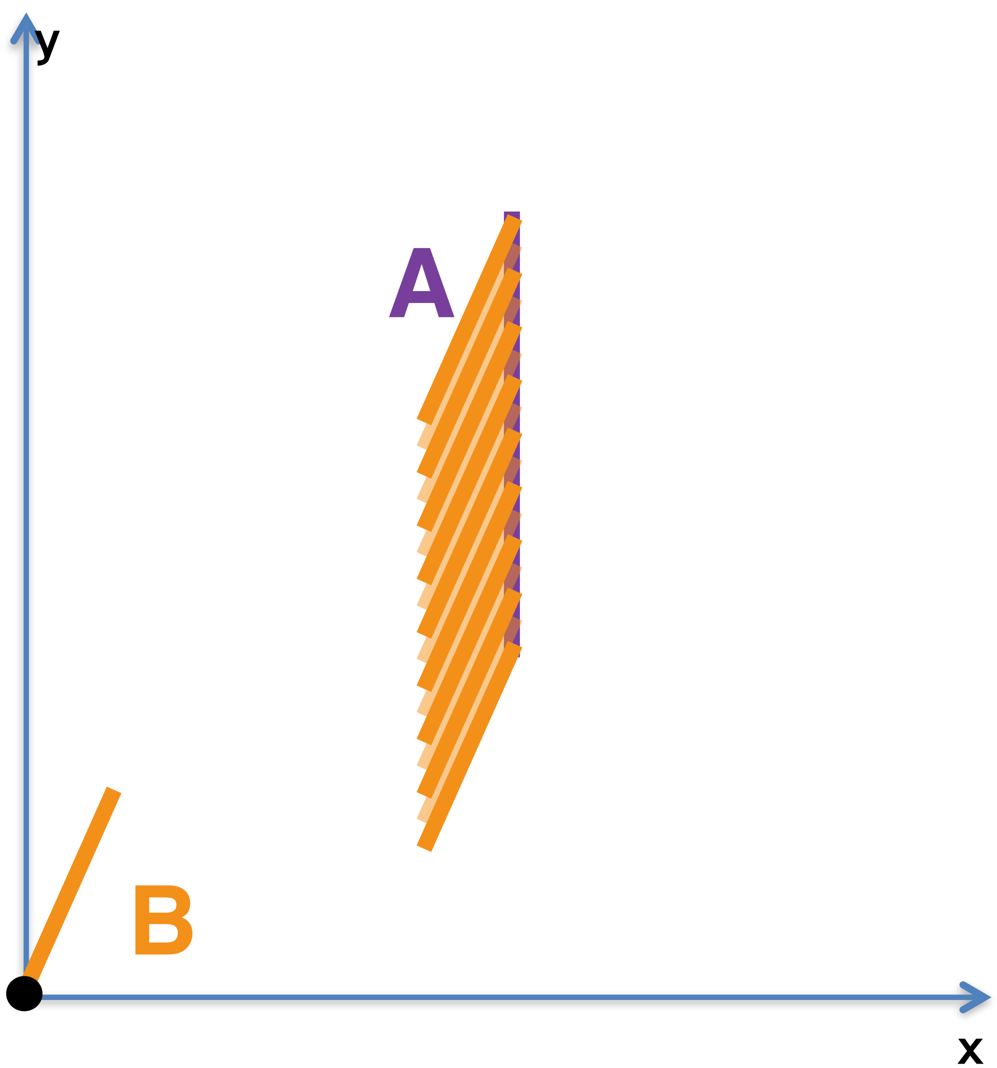
$$A \ominus B = \{a - b \mid a \in A, b \in B\}$$



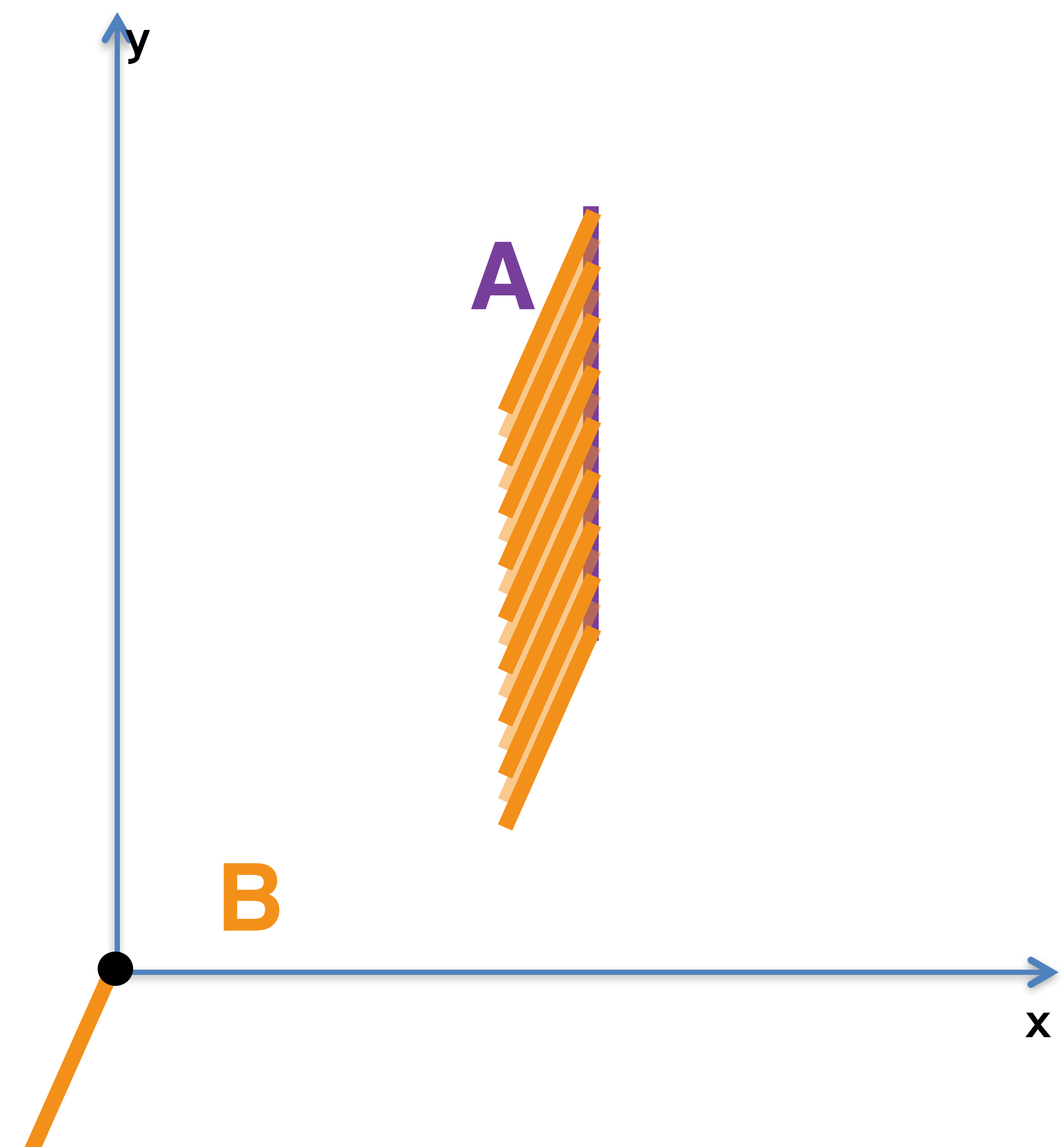
Minkowski Difference

- Difference is equivalent to sum with <u>reflected</u> B

$$A \ominus B = \{a - b \mid a \in A, b \in B\}$$



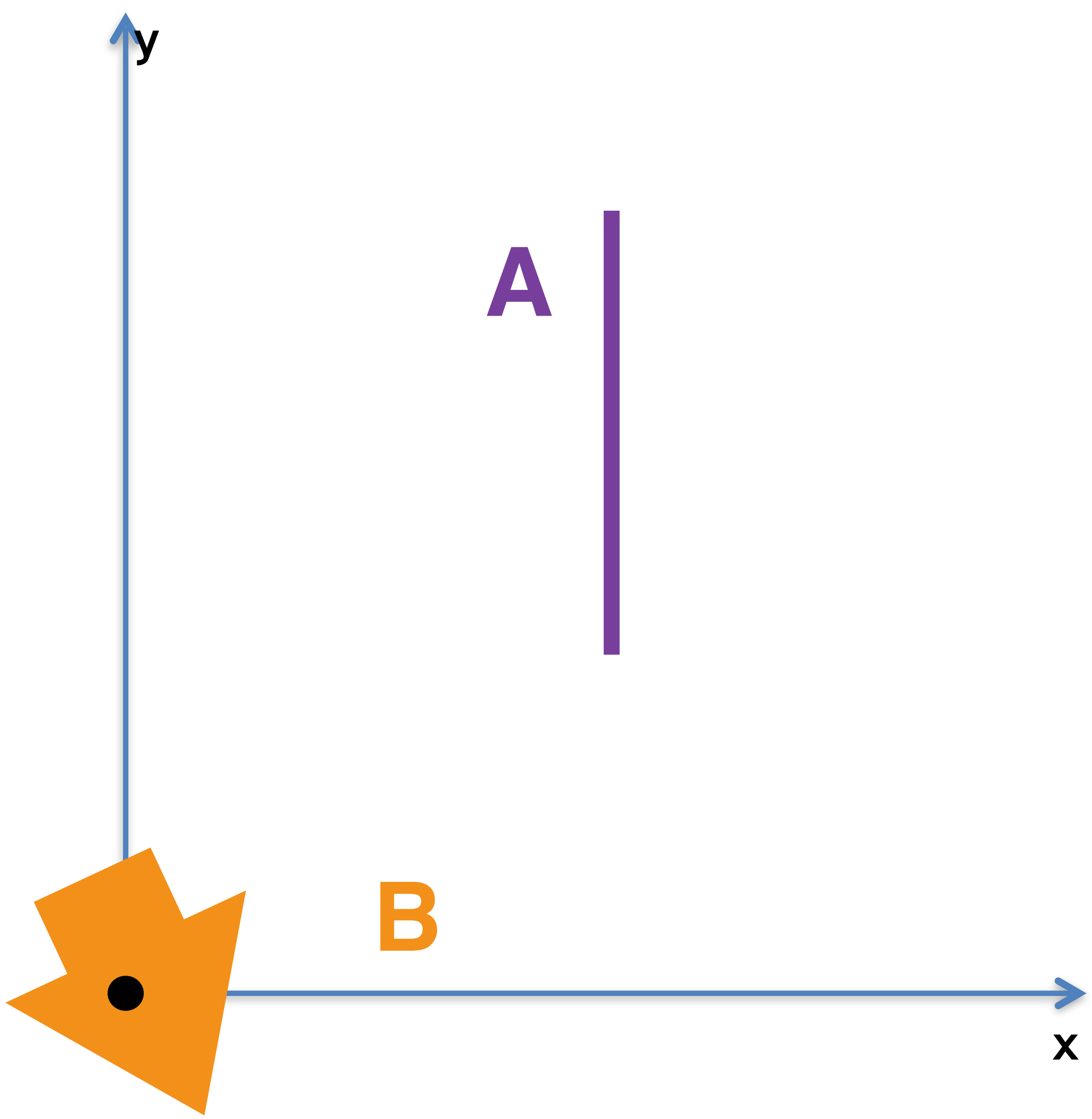
$$A \oplus B = \{a + b \mid a \in A, b \in B\}$$



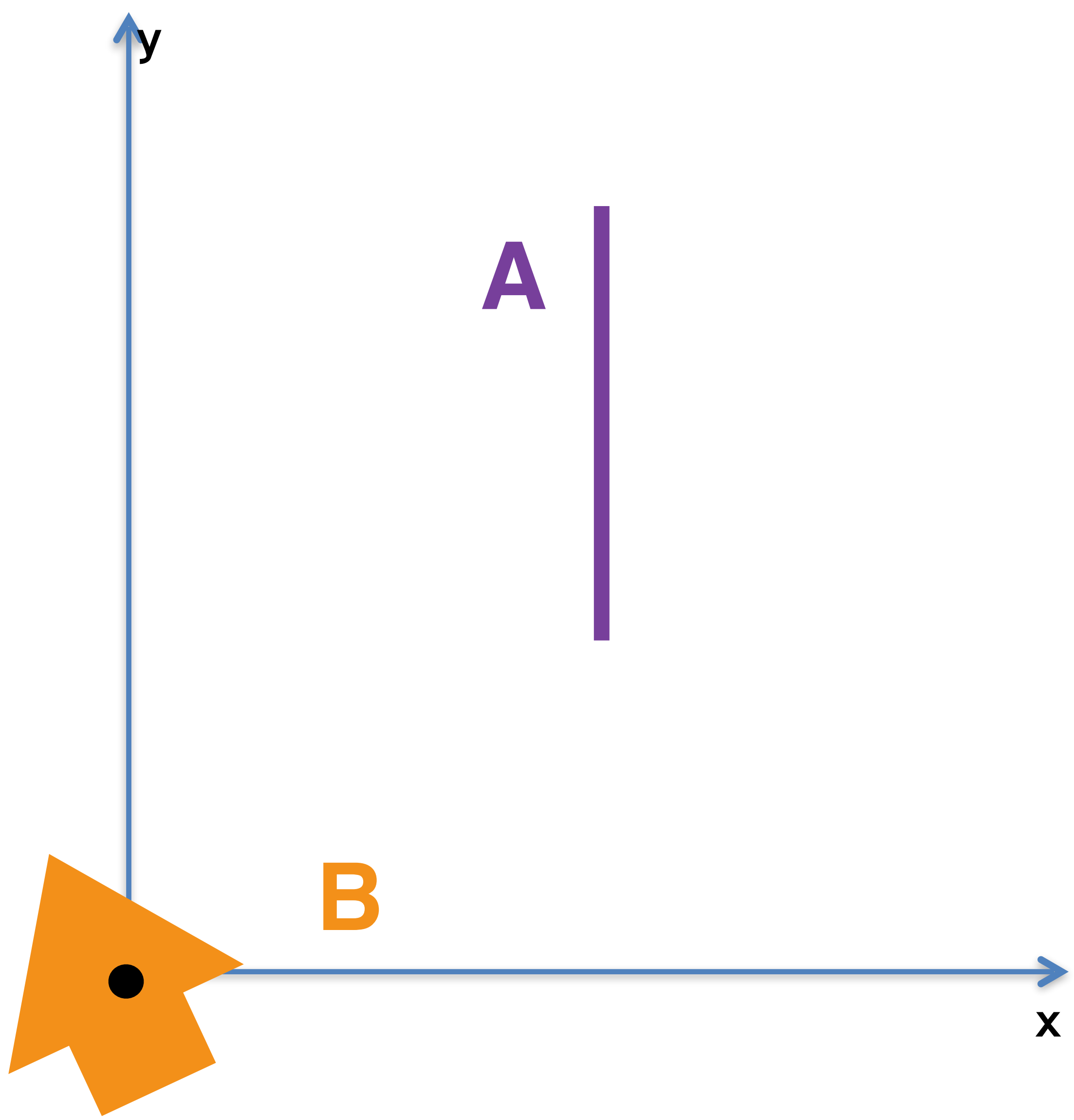
Minkowski Difference

- Difference is equivalent to sum with <u>reflected</u> B

$$A \ominus B = \{a - b \mid a \in A, b \in B\}$$



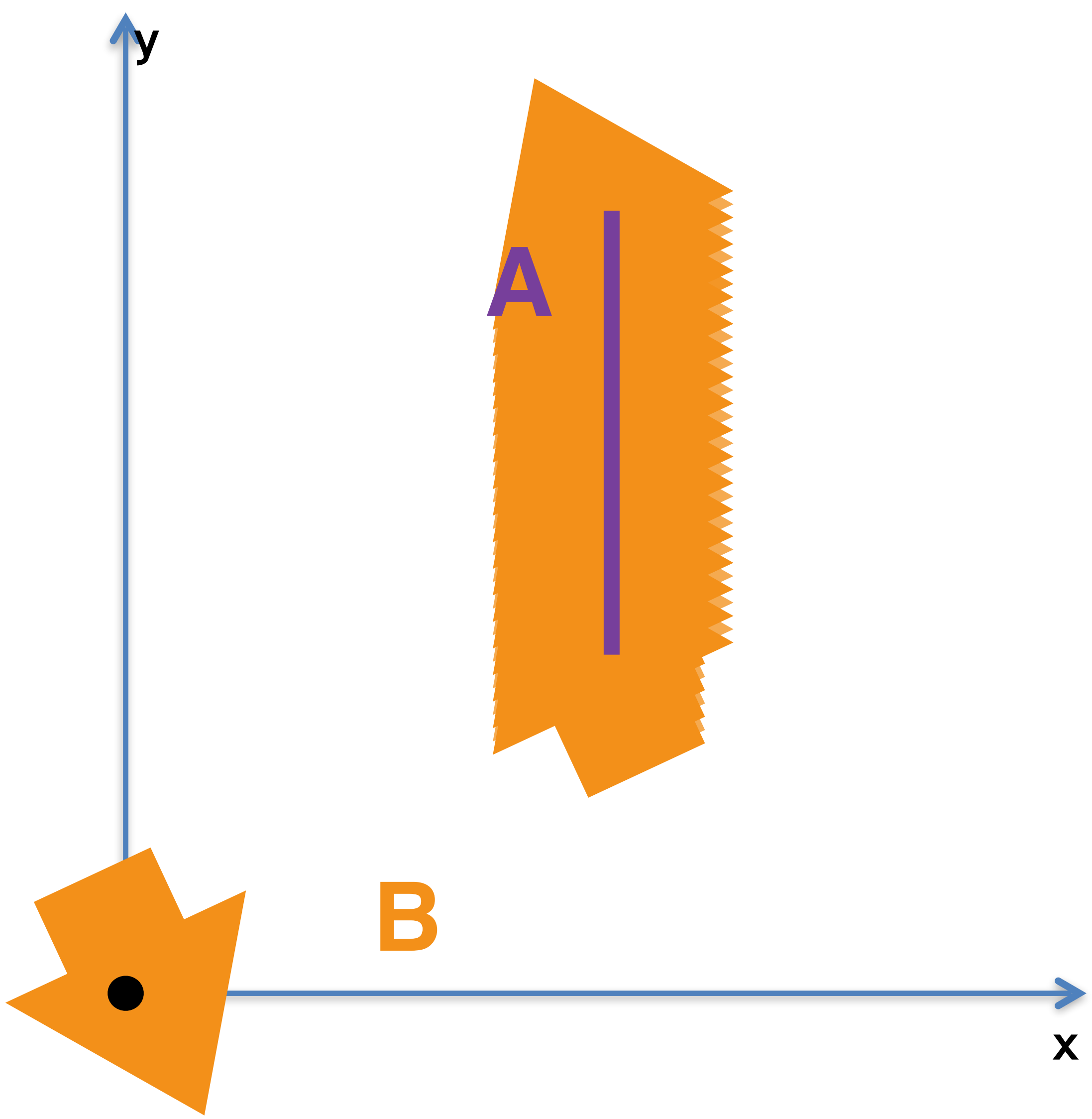
$$A \oplus B = \{a + b \mid a \in A, b \in B\}$$



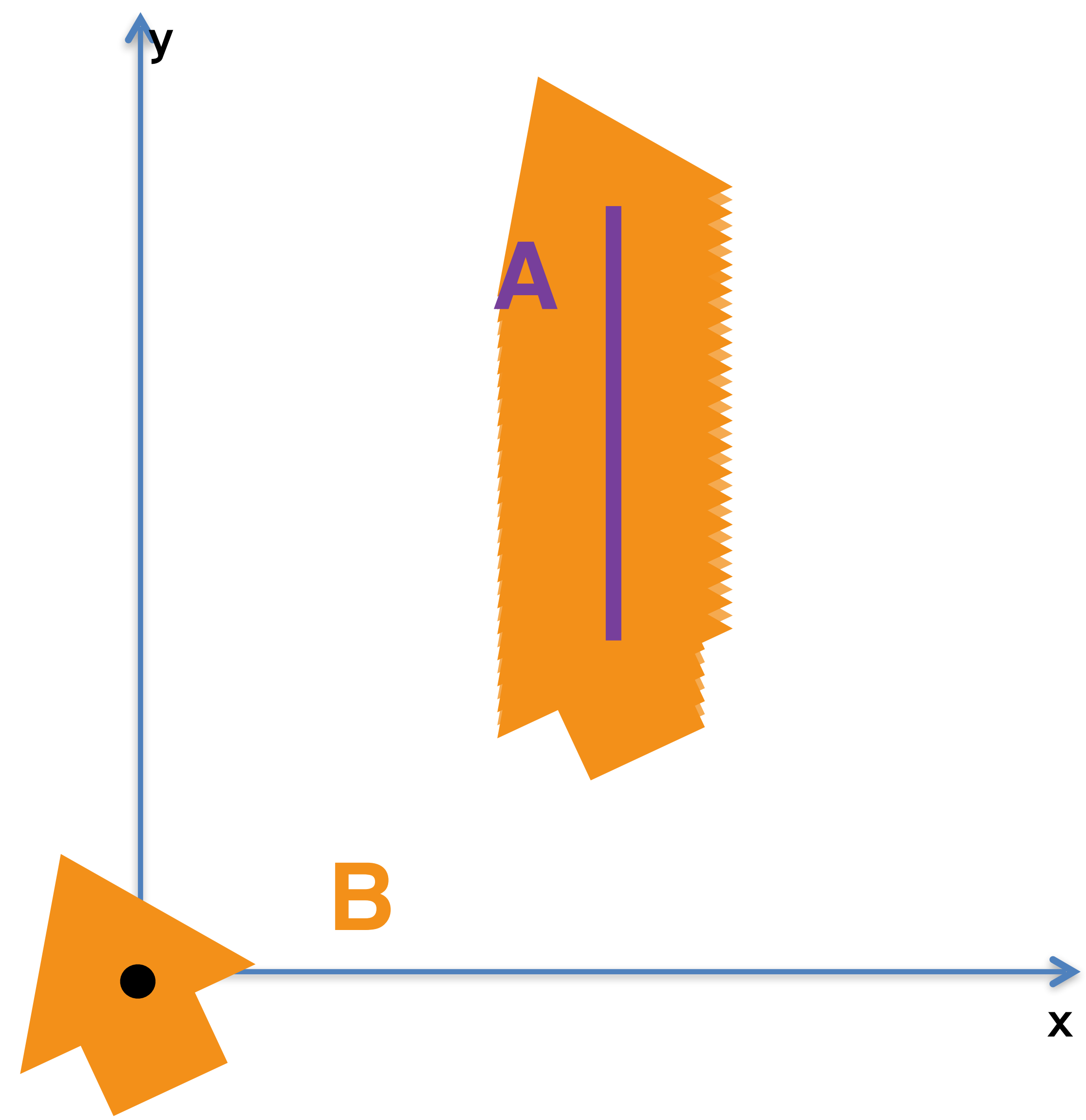
Minkowski Difference

- Difference is equivalent to sum with <u>reflected</u> B

$$A \ominus B = \{a - b \mid a \in A, b \in B\}$$



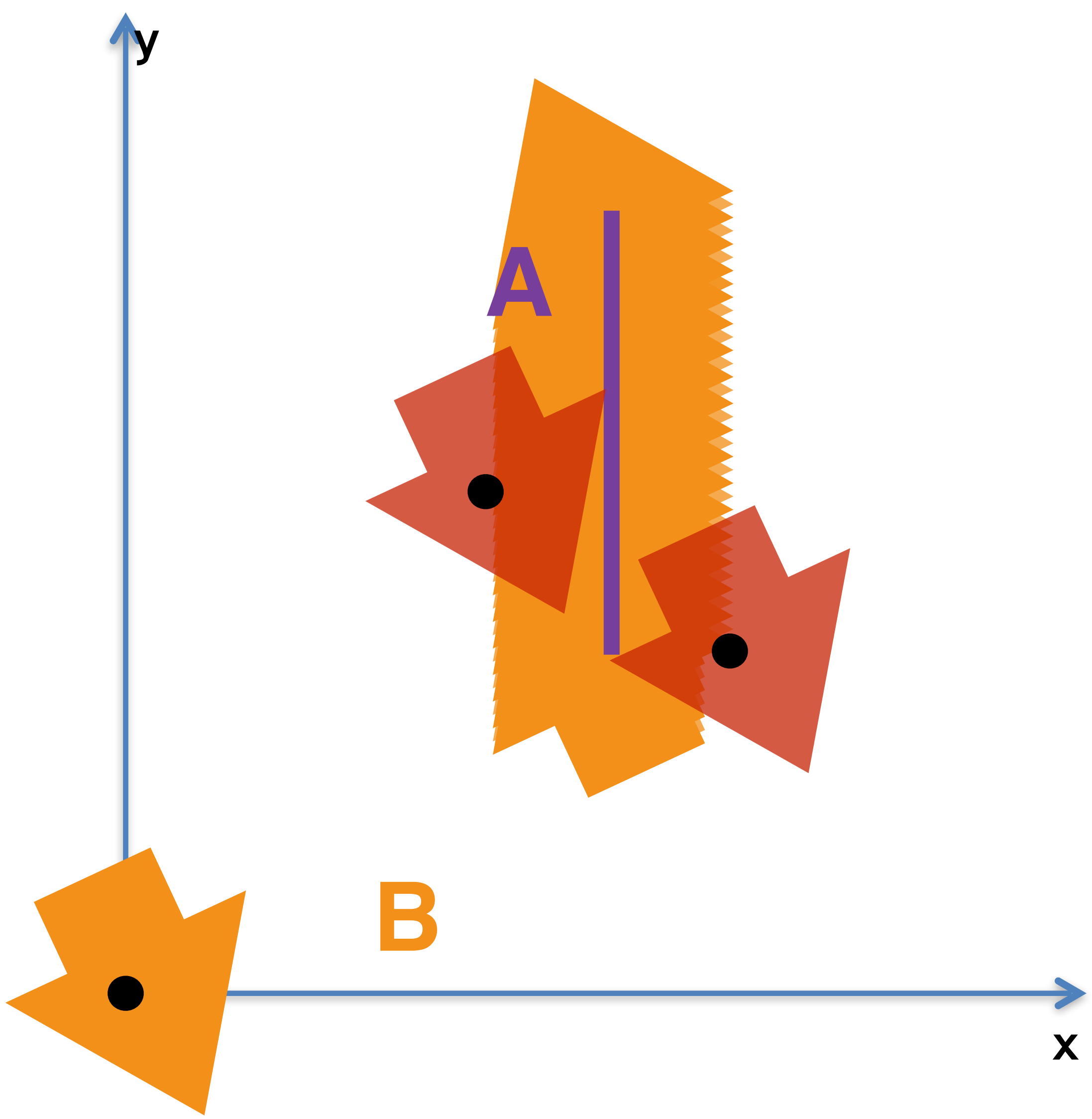
$$A \oplus B = \{a + b \mid a \in A, b \in B\}$$



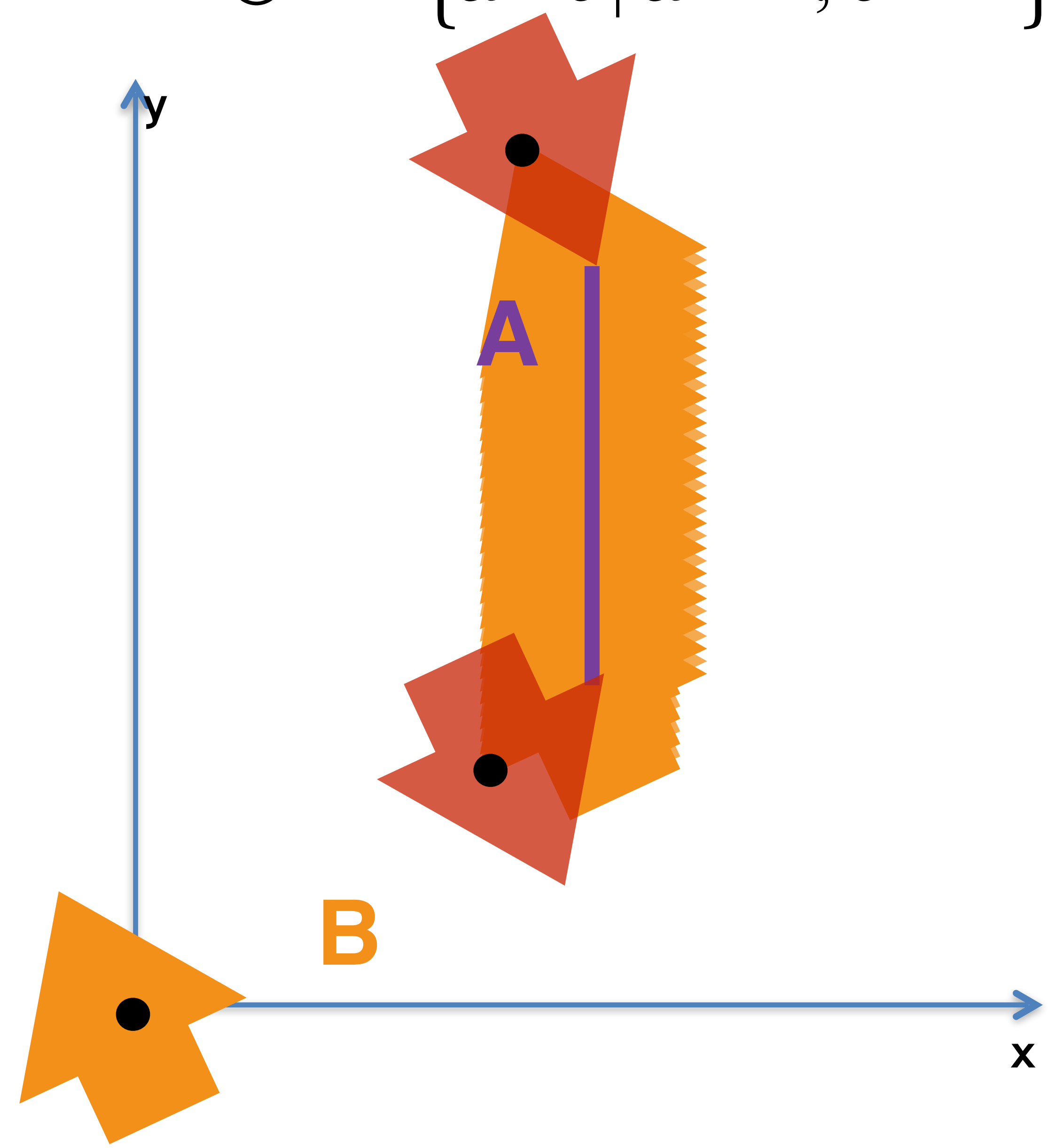
Minkowski Difference

- Difference is equivalent to sum with <u>reflected</u> B

$$A \ominus B = \{a - b \mid a \in A, b \in B\}$$

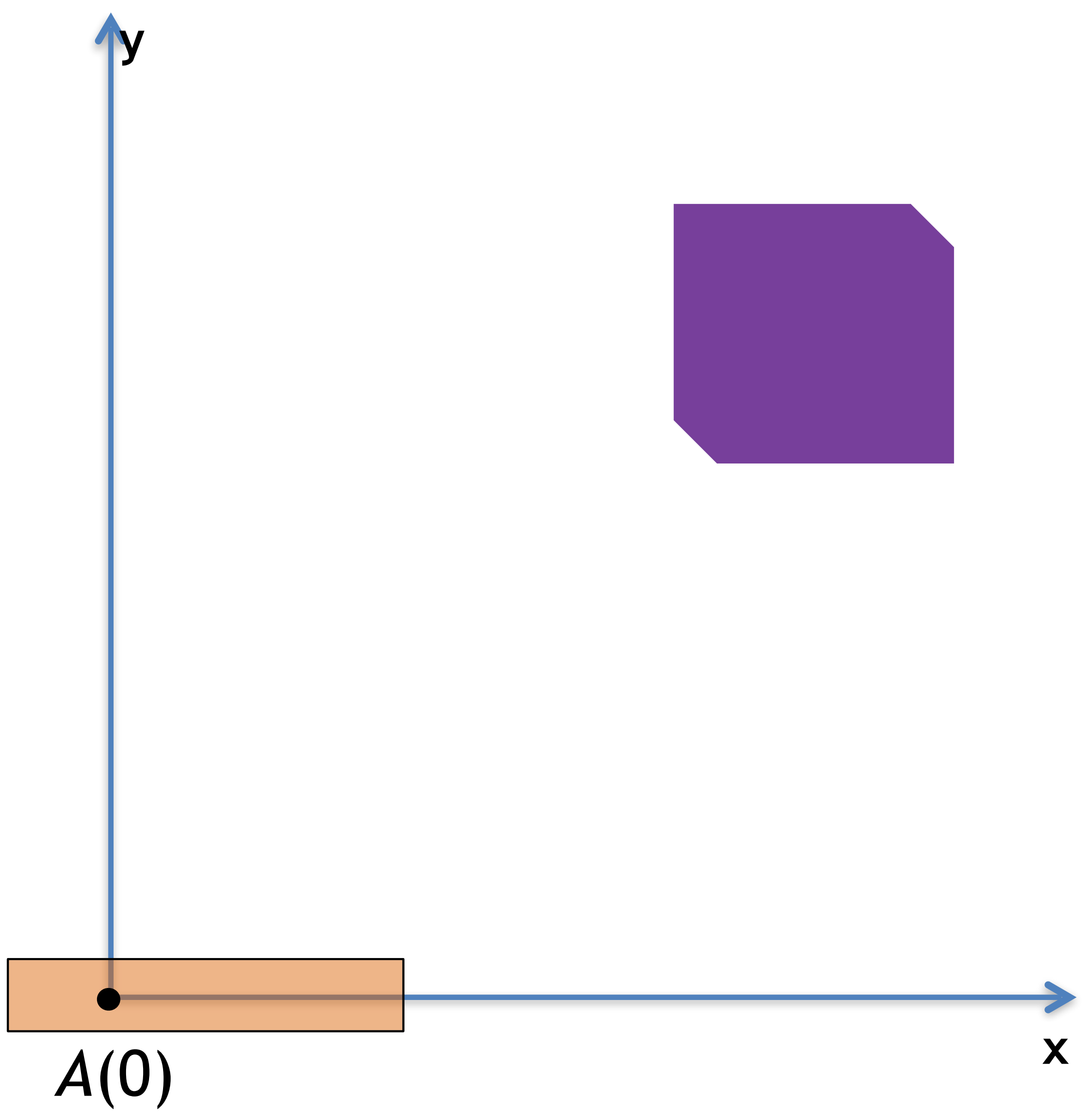


$$A \oplus B = \{a + b \mid a \in A, b \in B\}$$



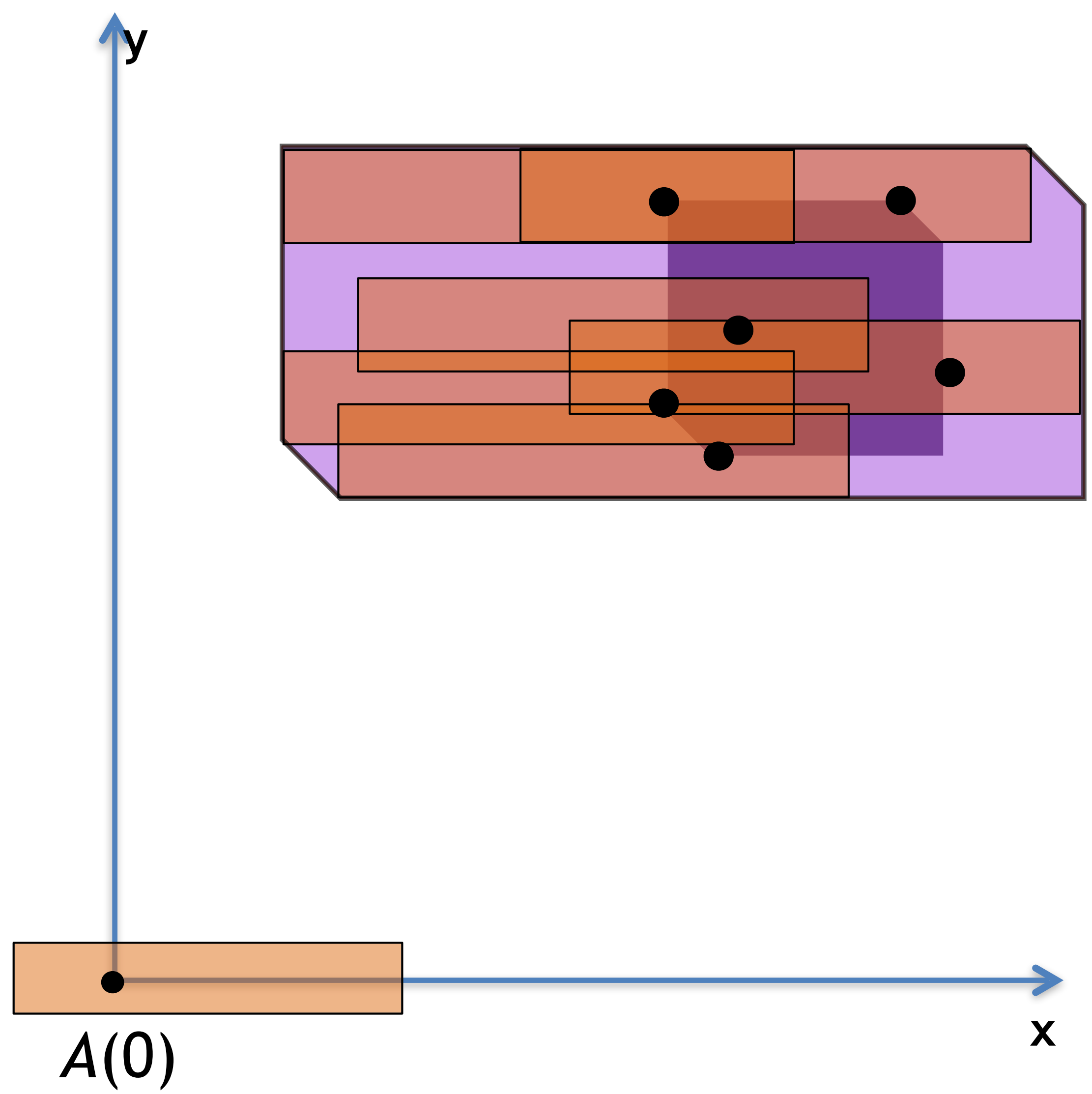
Minkowski Difference

$$A \ominus B = \{a - b \mid a \in A, b \in B\}$$



Minkowski Difference

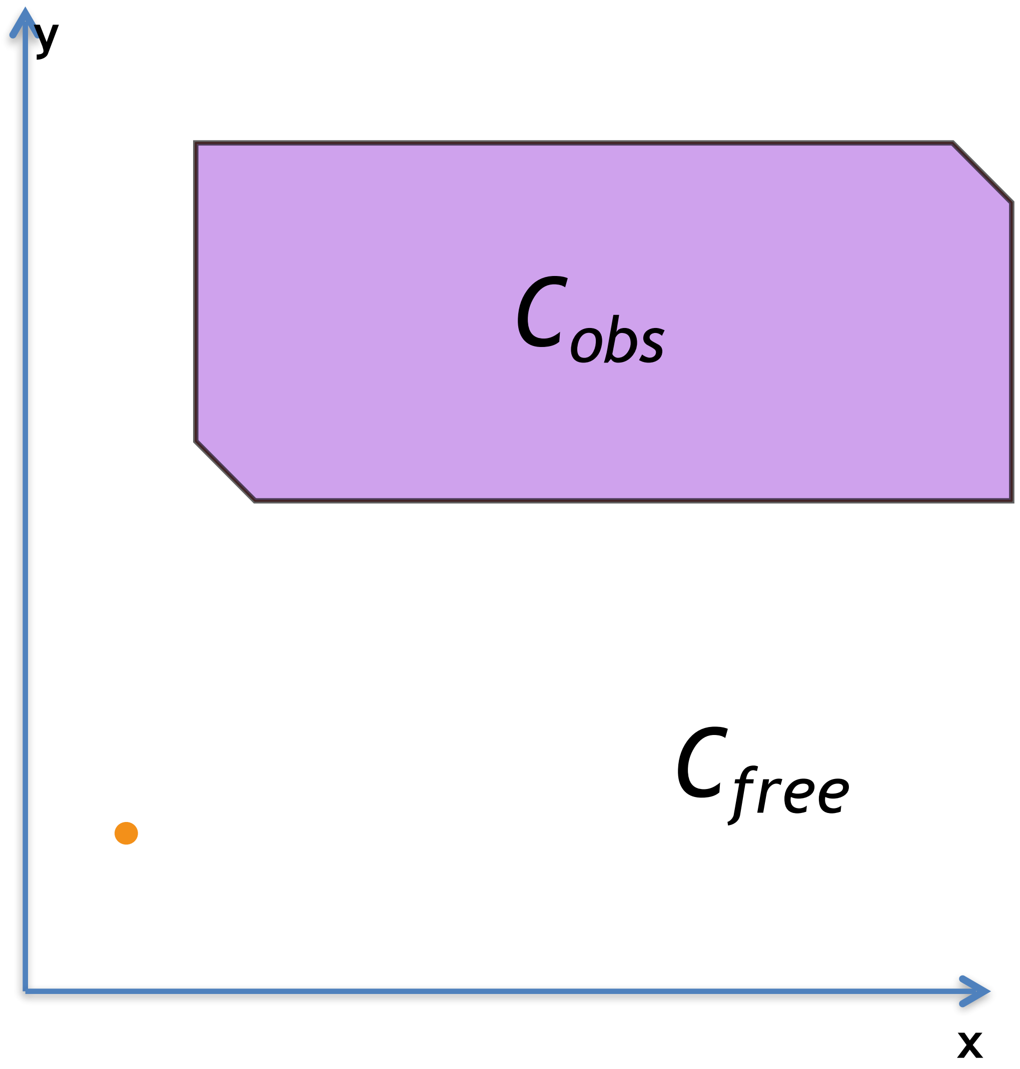
$$A \ominus B = \{a - b \mid a \in A, b \in B\}$$



Sweeps reflected shape of B

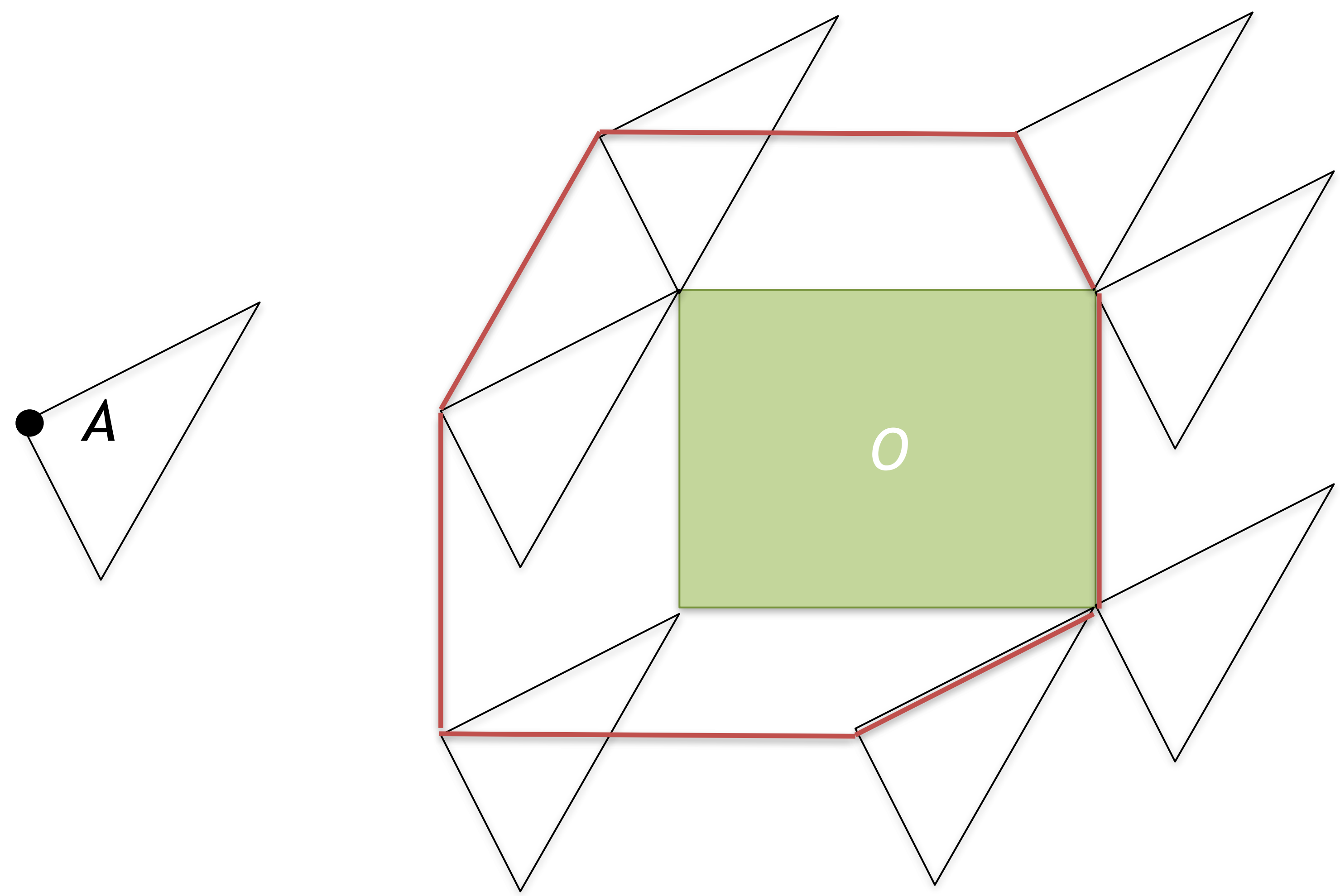
Minkowski Difference

$$A \ominus B = \{a - b \mid a \in A, b \in B\}$$



Star Algorithm

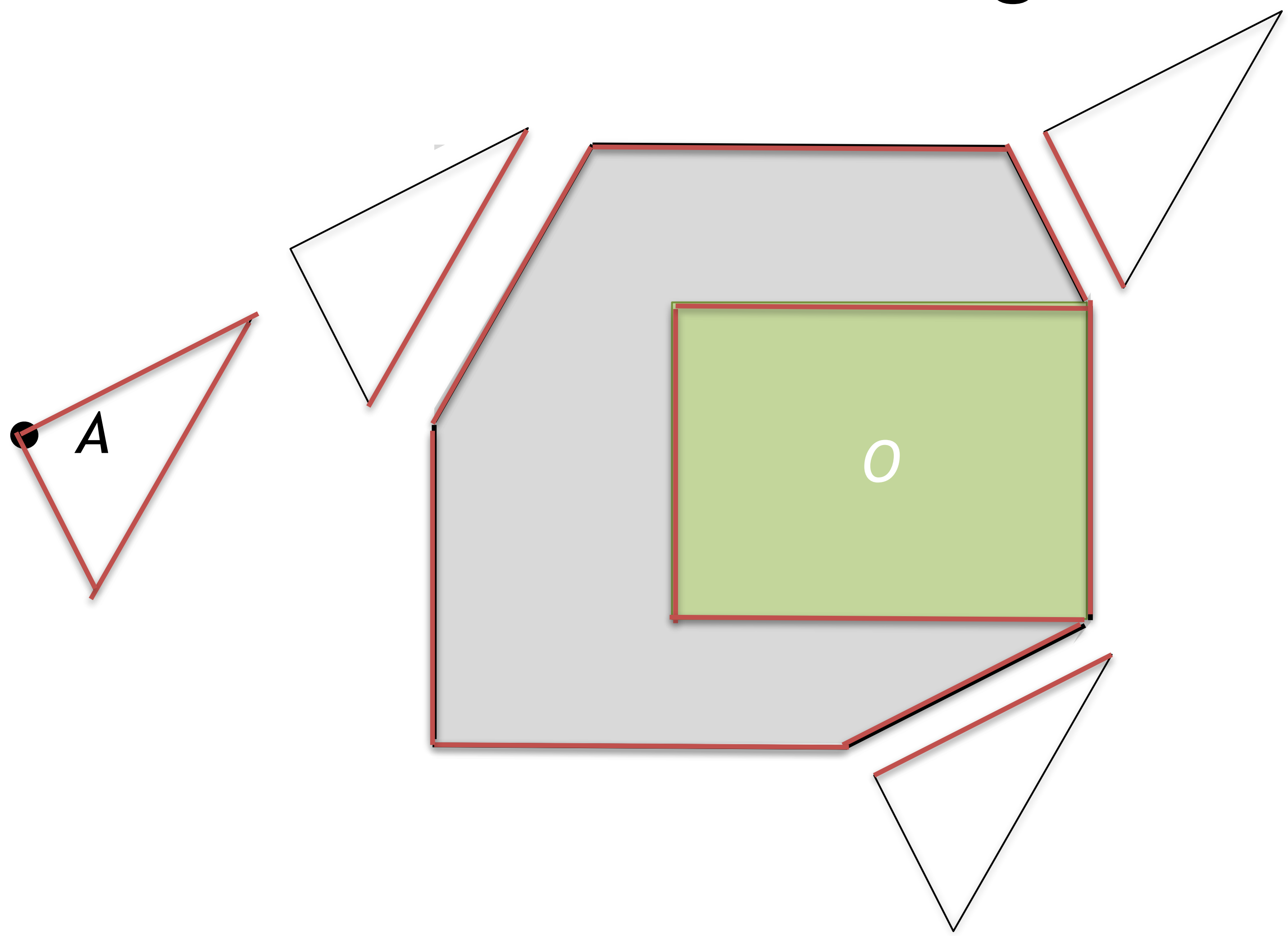
- 2D workspace and only translation
- Robot and obstacles are convex polygons



- C-space obstacle C_{obs} is also convex
- How to efficiently compute shape of c-space obstacle?

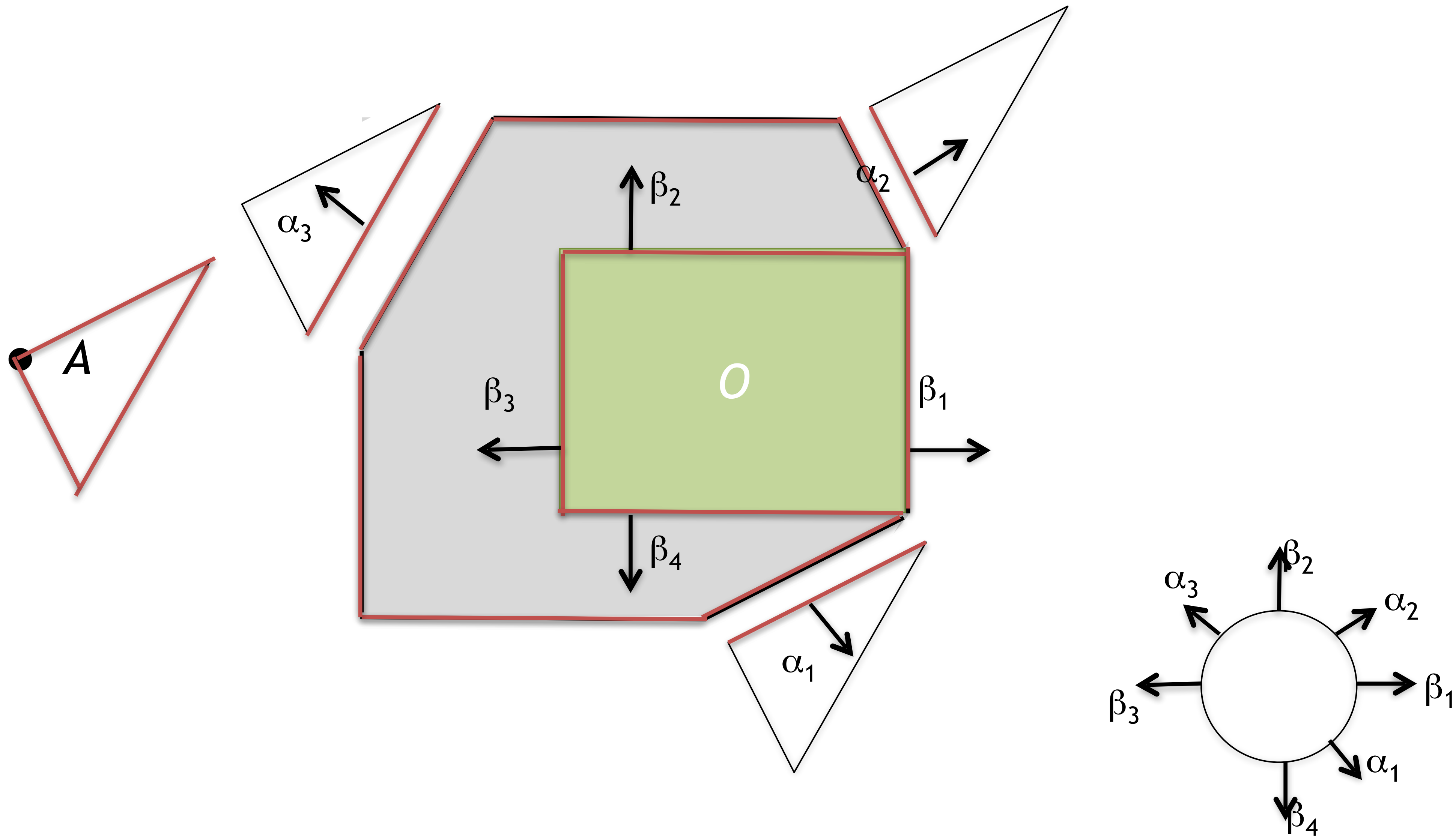
Star Algorithm

- Sides of C_{obs} correspond to sides of obstacle and robot
- In which order are the sides arranged though?



Star Algorithm

- Sides are arranged according to their normals as shown
    - Robot normals point inwards, obstacle normals point outwards

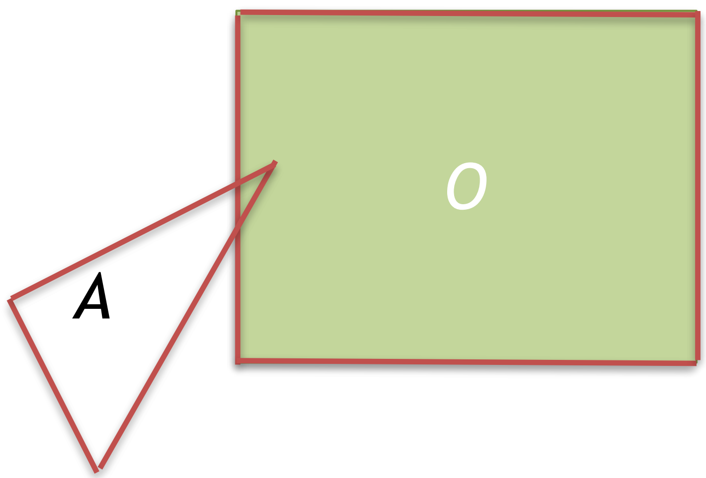


- Convex c-space obstacle represented by half-planes

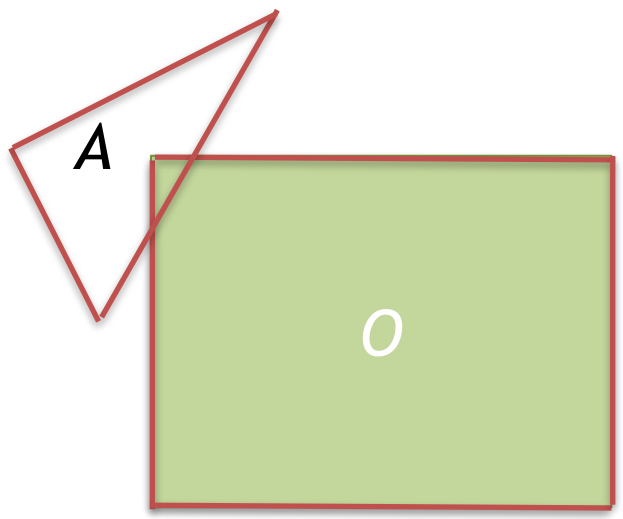
Star Algorithm

- Two types of contacts:

Vertex-Edge



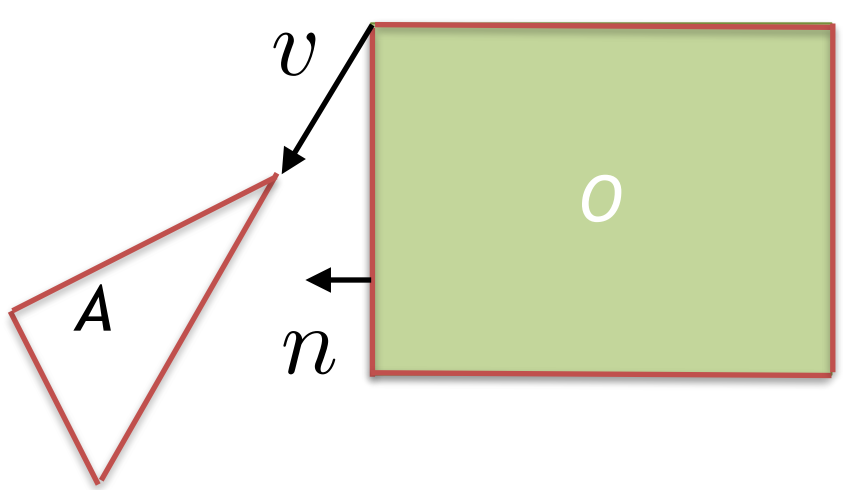
Edge-Vertex



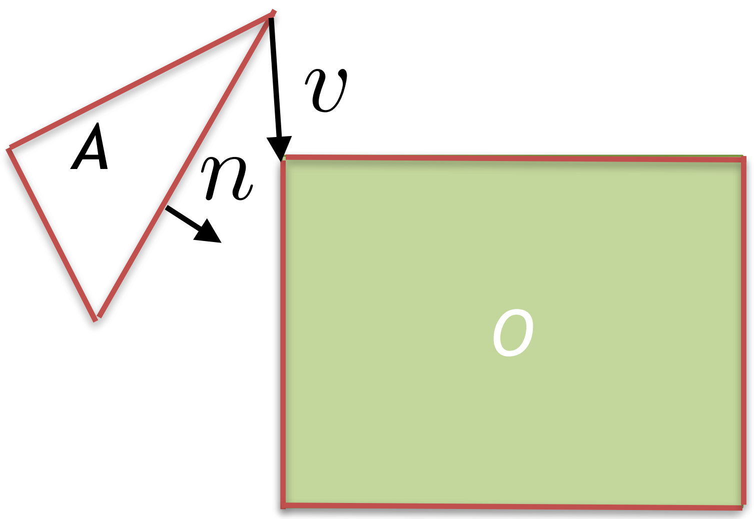
Star Algorithm

- Two types of contacts:

Vertex-Edge



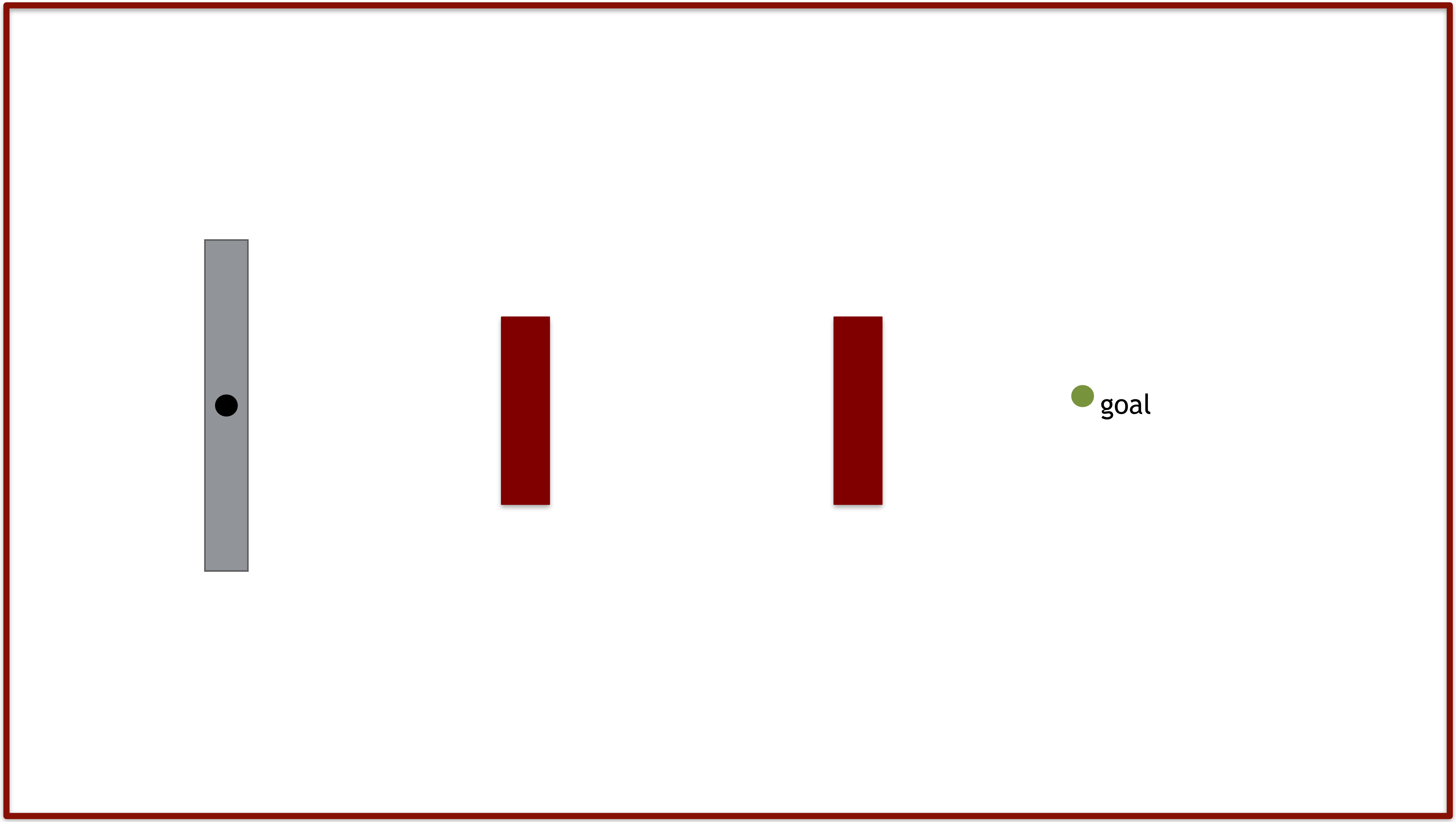
Edge-Vertex



- Contact corresponds to $n^T v(x,y) = 0$ line in c-space
- Half-plane defined by $H = \{(x,y) | n^T v(x,y) \leq 0\}$

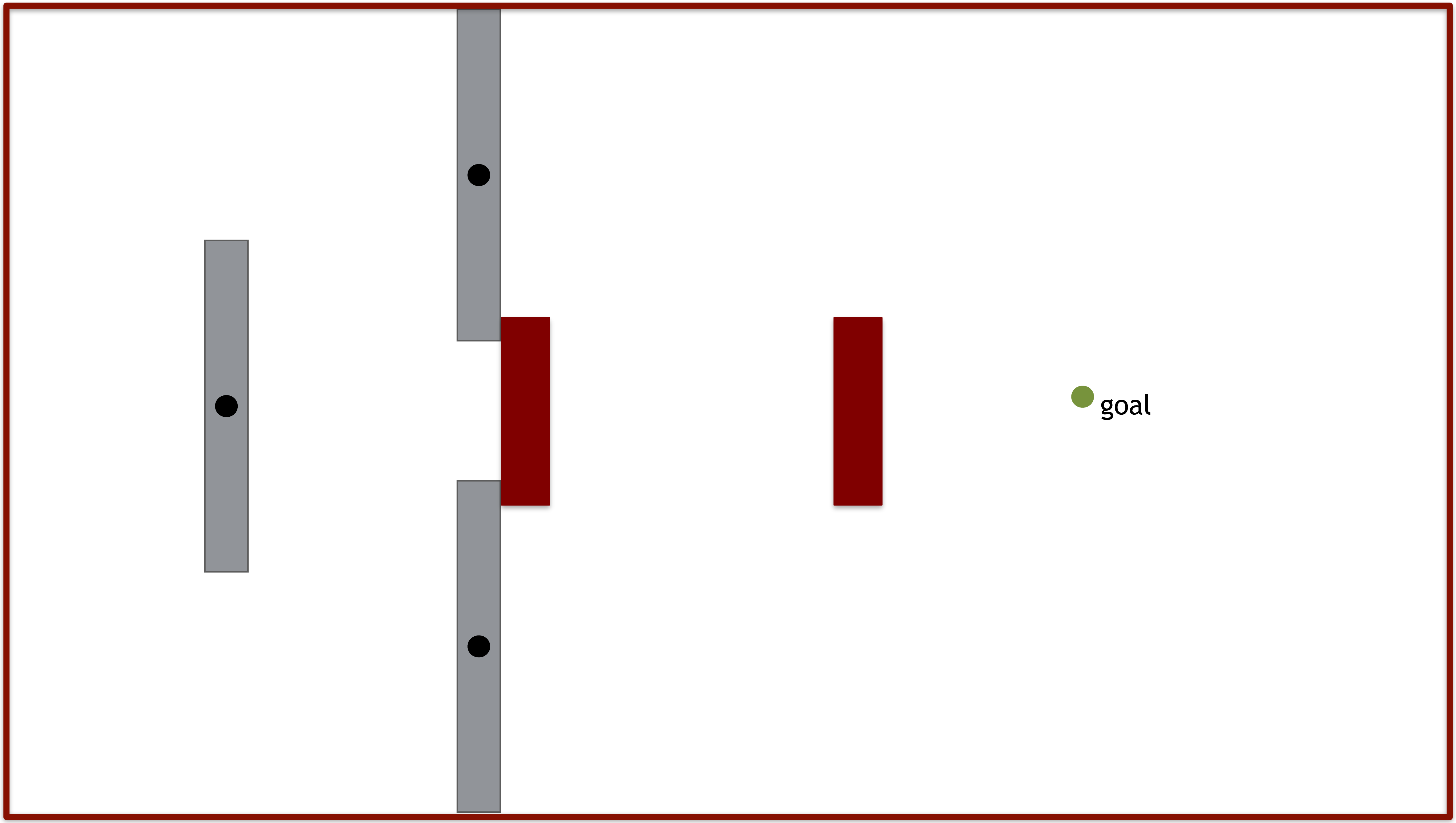
Orientation

Consider the following piano movers' problem



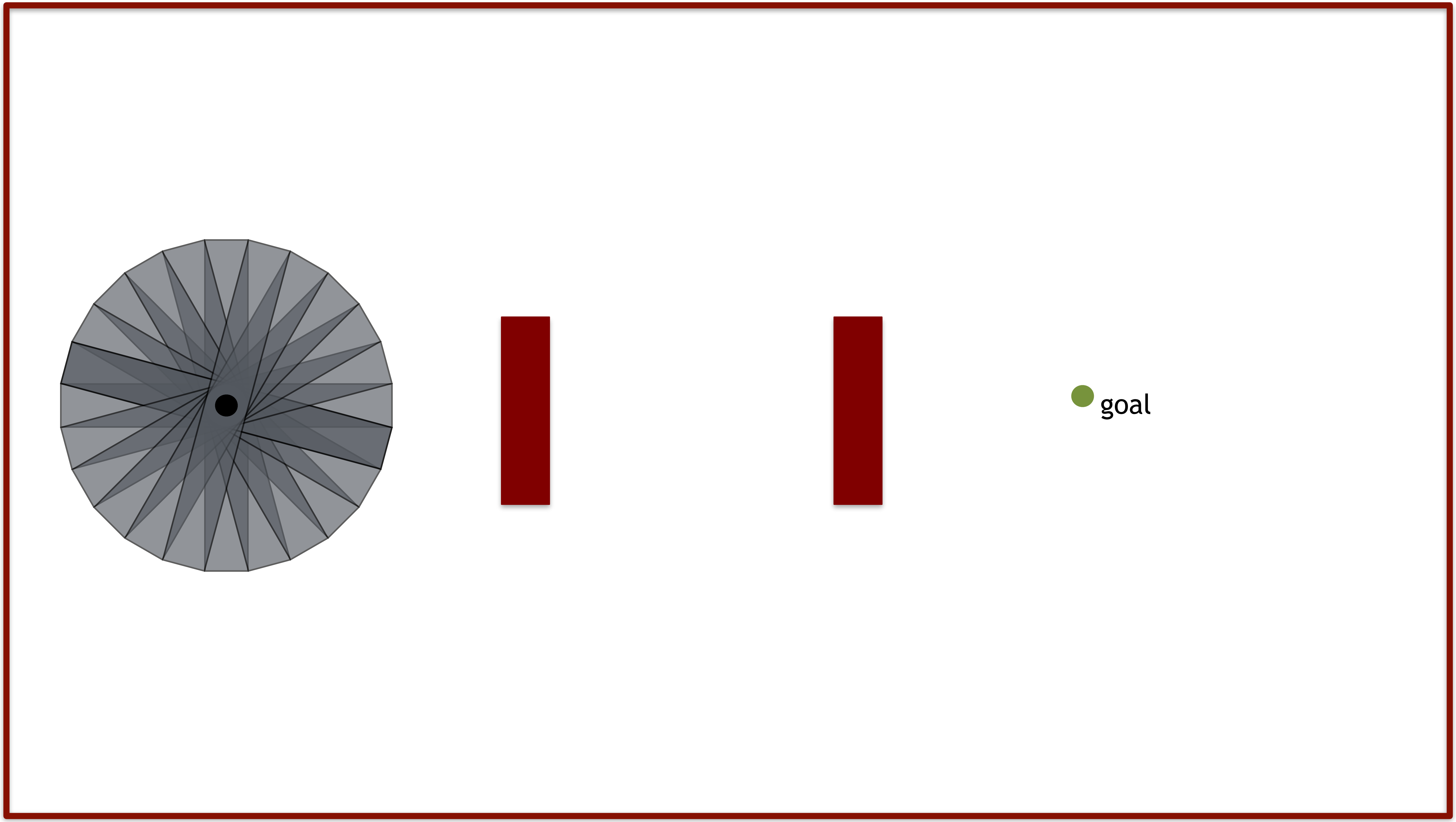
Orientation

The path is blocked when using the current orientation



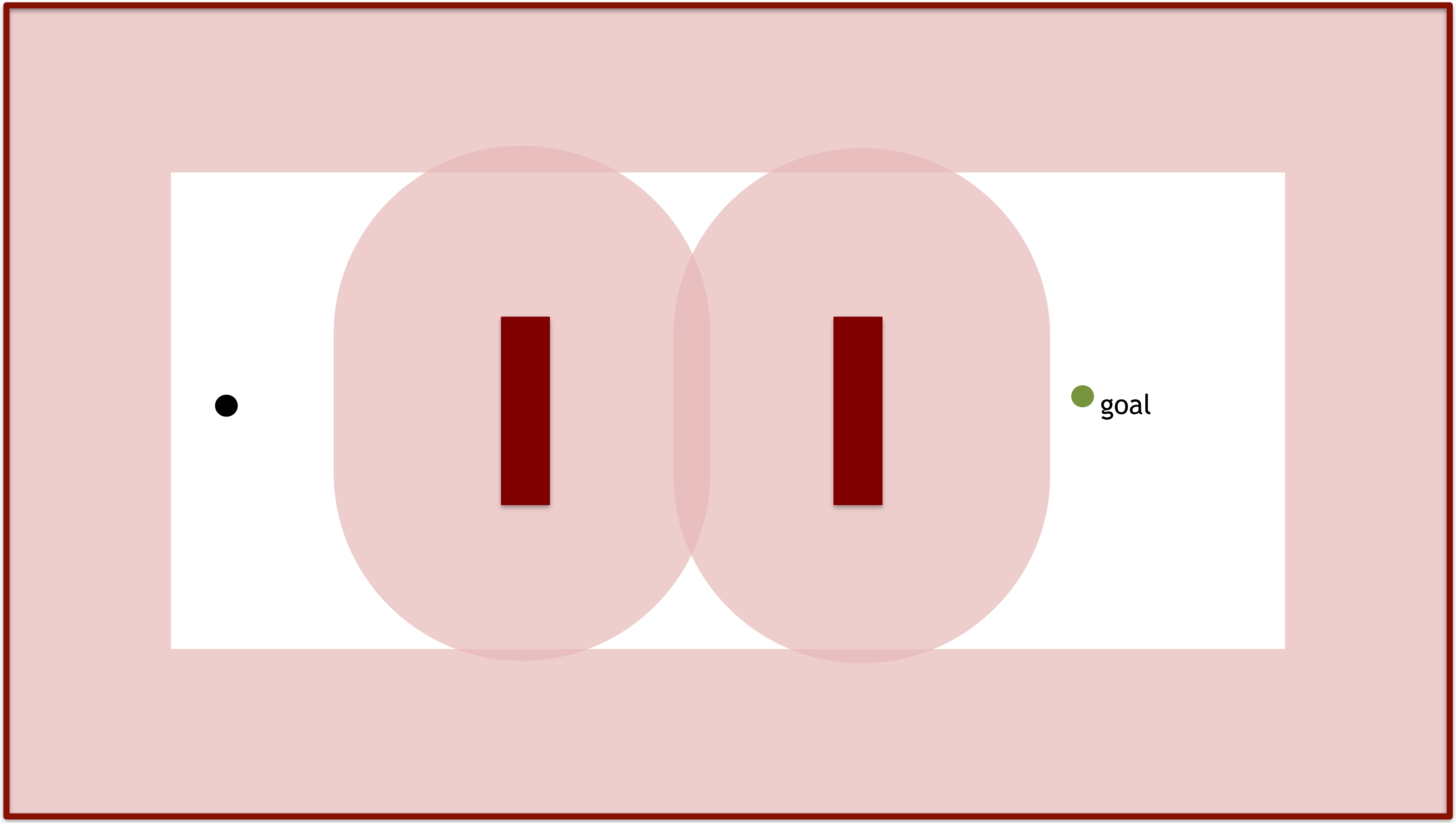
Orientation

A very conservative naive approach would cover all angles



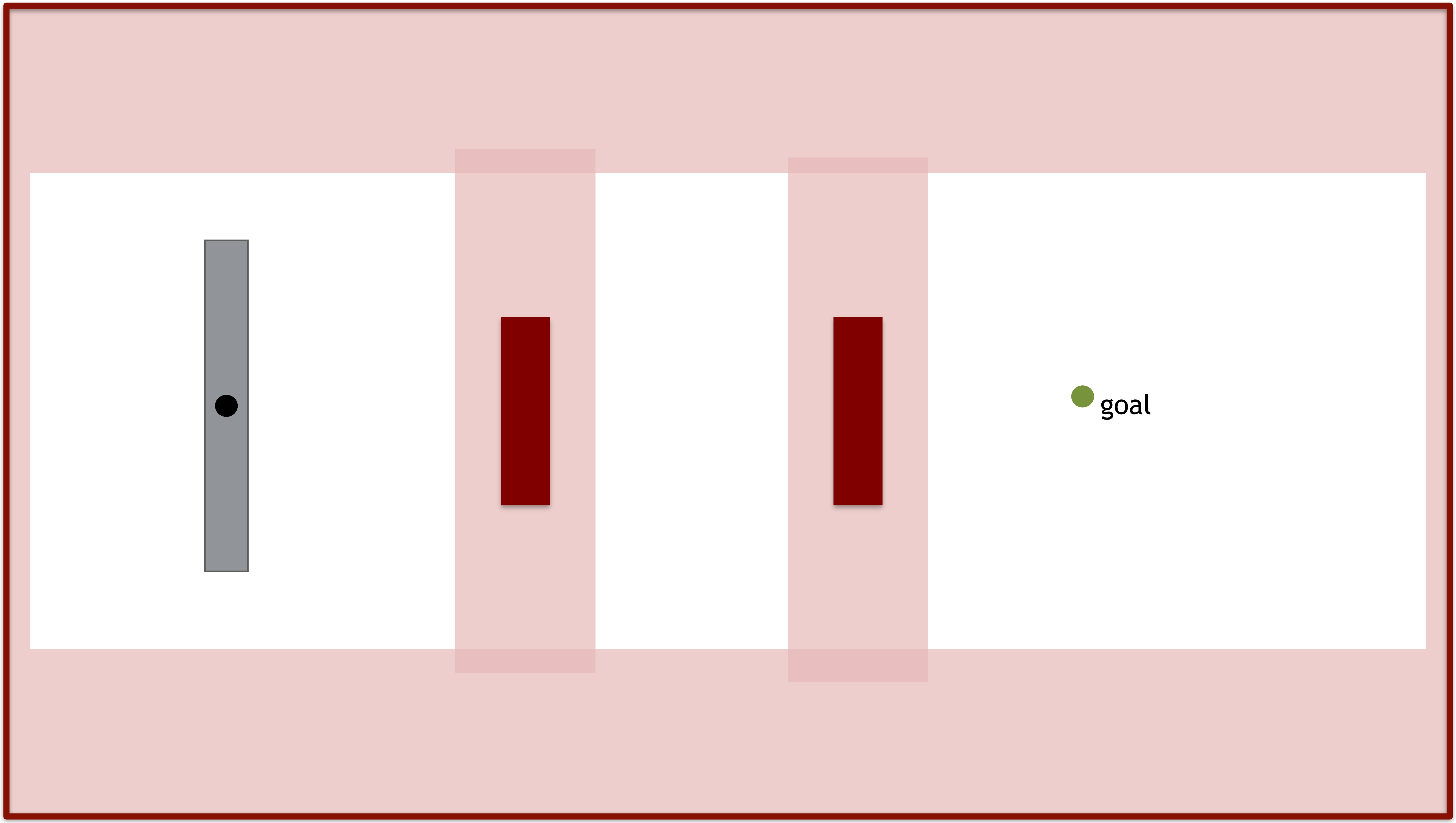
Orientation

Need to consider different obstacles for individual angles



Orientation

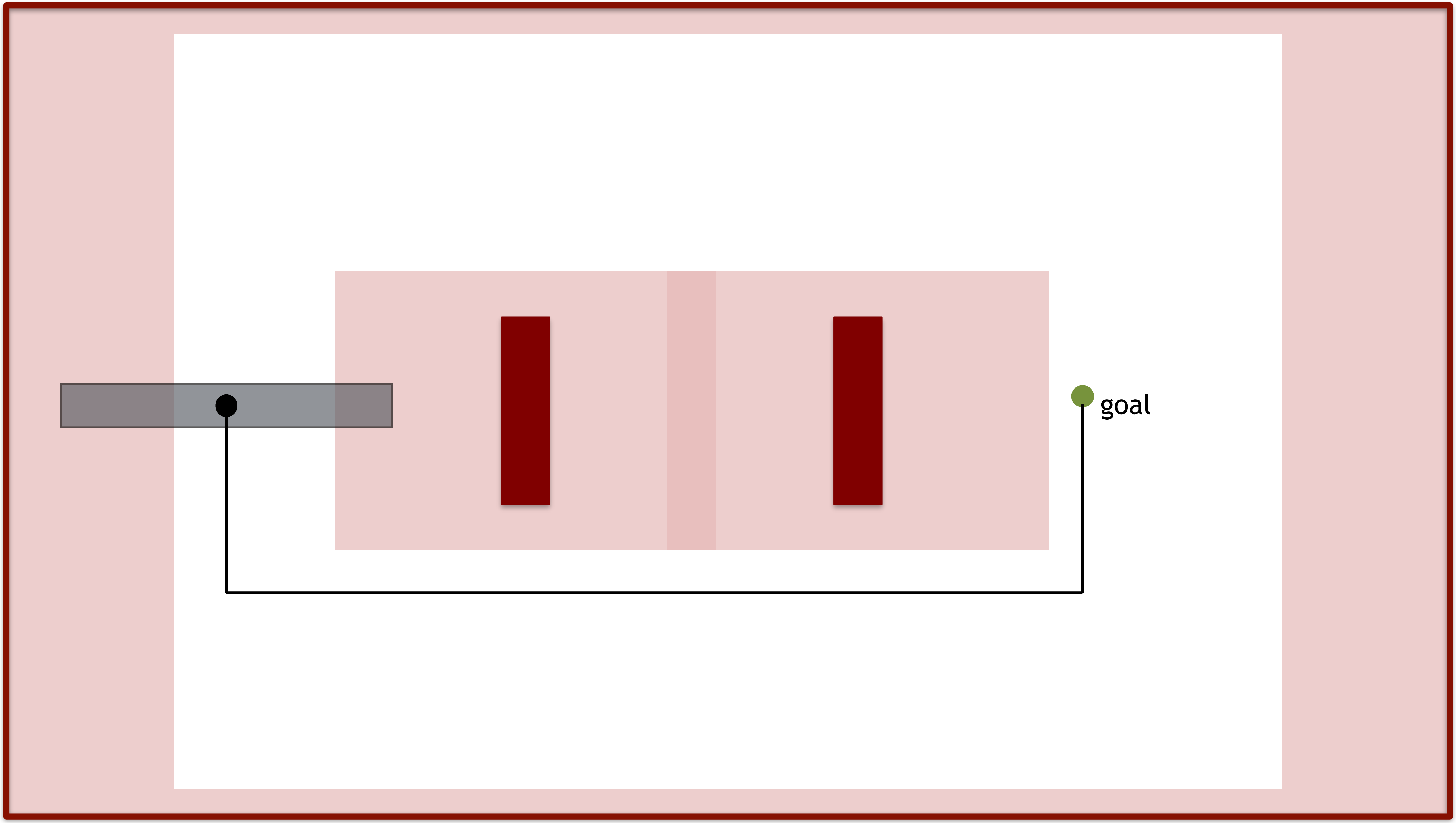
Need to consider different obstacles for individual angles



Note how the c-space obstacles change with orientation

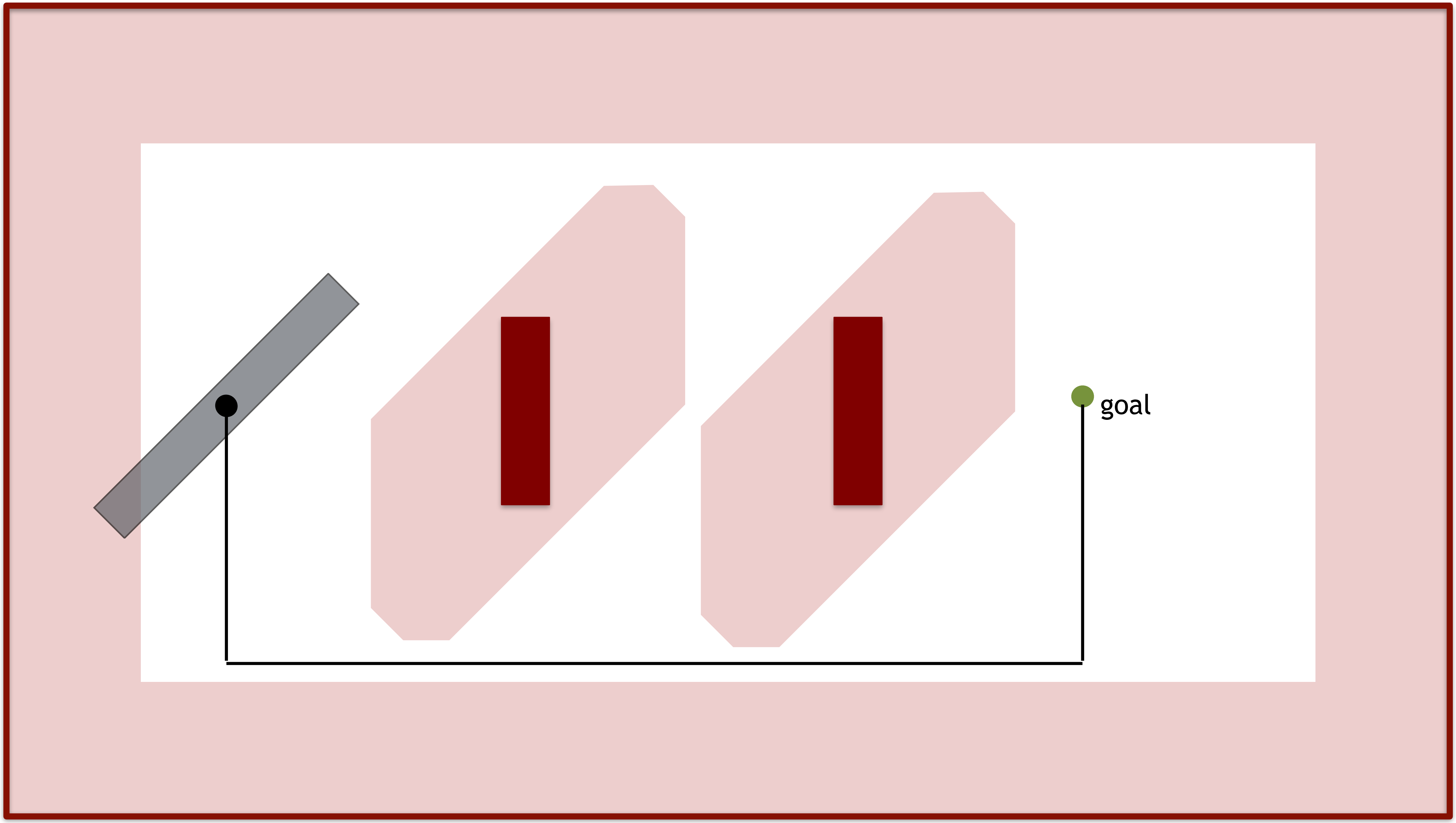
Orientation

Need to consider different obstacles for individual angles



Orientation

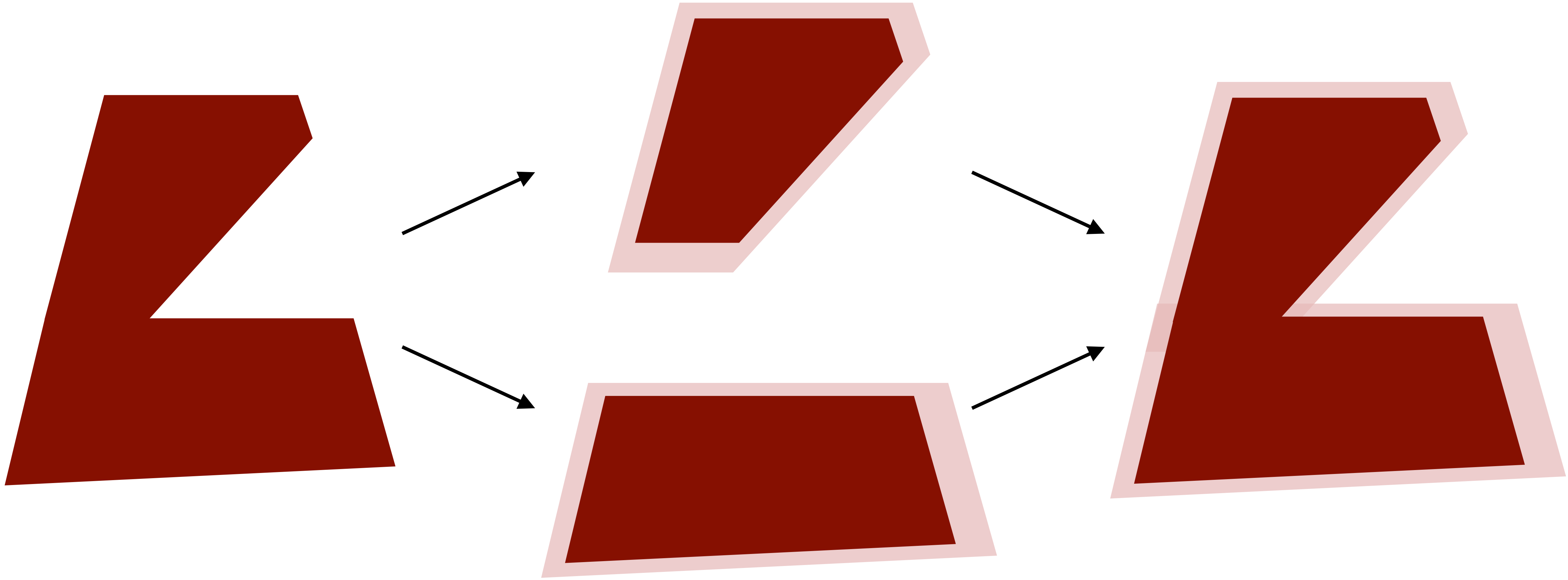
Need to consider different obstacles for individual angles



Note how the c-space obstacles change with orientation

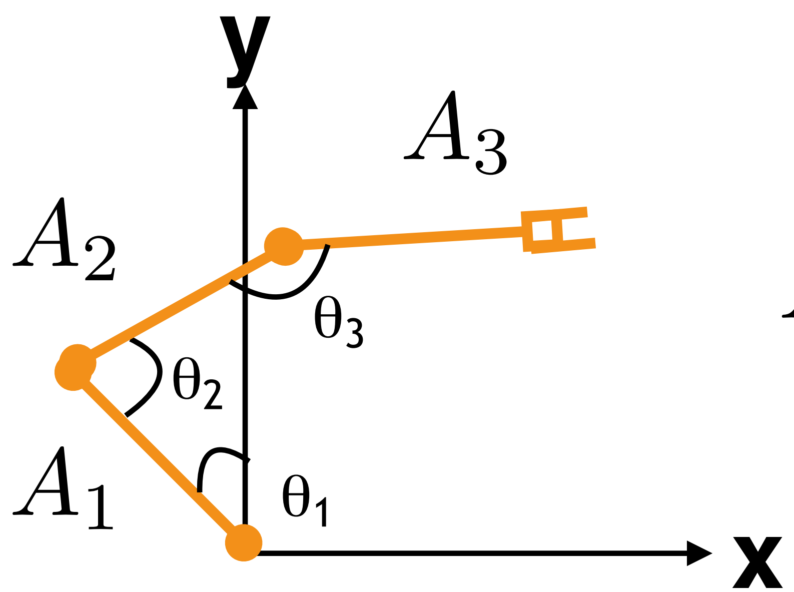
Non-Convex Obstacles

- Not all obstacles O will be convex, as we have assumed
- Compute C_{obs} as union of of convex components of O



Articulated Robots

- Articulated robots consist of multiple links
- Links can collide with each other leading to self-collision
- Consider each link independently

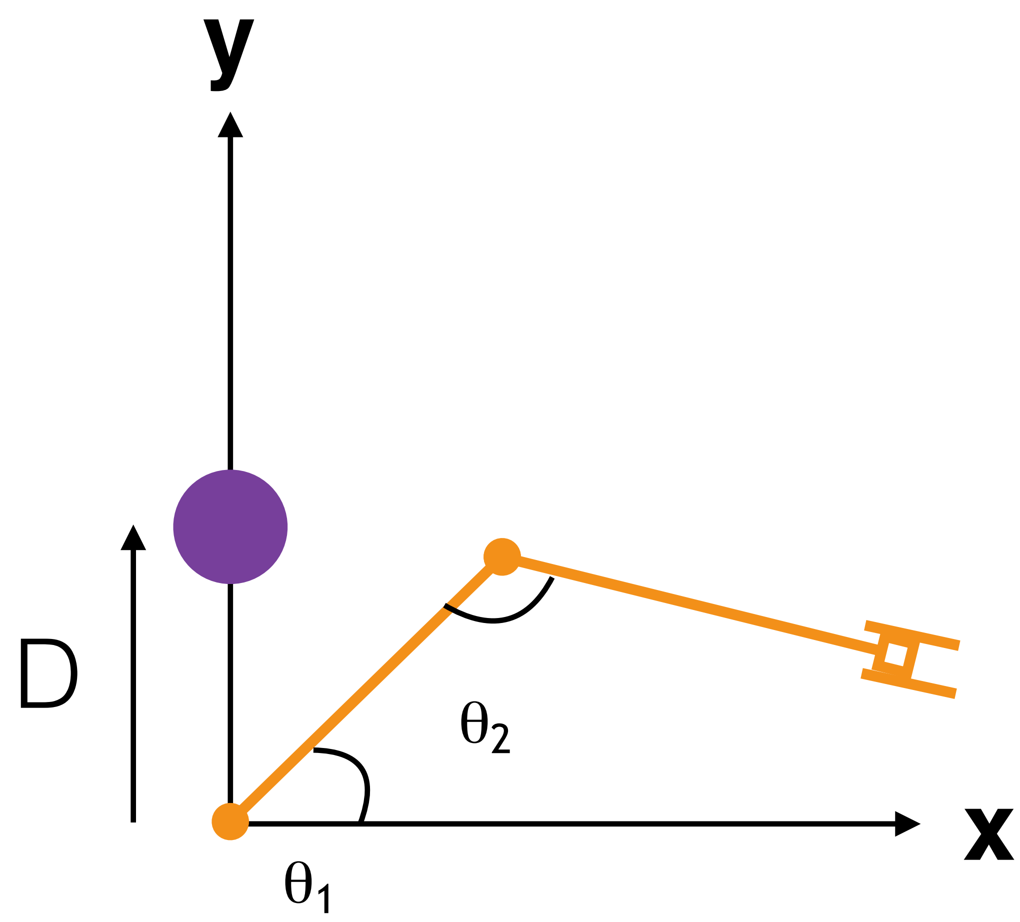


$$A = \{A_1, A_2, ..., A_m\}$$

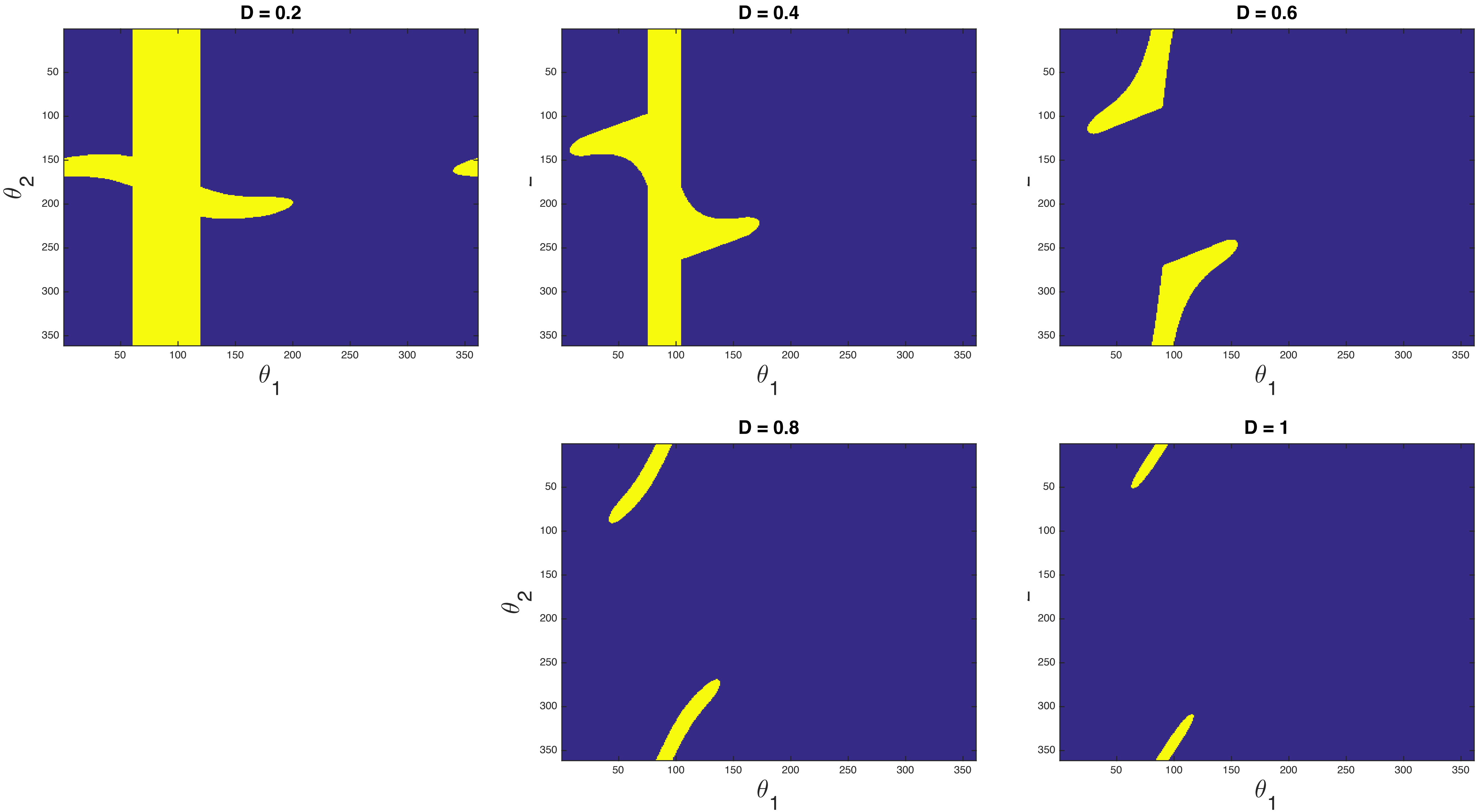
$$C_{obs} = \left(\bigcup_{i=1}^{m} \{q \in C \mid A_i(q) \cap O \neq \emptyset\} \right) \cup \left(\bigcup_{[i,j]\in P} \{q \in C \mid A_i(q) \cap A_j(q) \neq \emptyset\} \right)$$

$$\text{Collision pairs: } (i, j) \in P \forall i \neq j$$

Example of Obstacle for Articulated Robot



Example of Obstacle for Articulated Robot



Questions?