Obstacle Representation in Configuration Space

Pradipta Biswas, PhD (Cantab)
Associate Professor
Indian Institute of Science

https://cambum.net/

Content

Related concepts

Open Set

Manifold

Convex set

Minkowski Sum and Difference

- Configuration Space
  - Definitions
  - Examples
- Obstacle Representations
  - Challenges
  - Requirement
- Minkowski Sum
  - Definition
  - Examples
- 1D Case
- 2D Case
  - Star Algorithm
  - Contour Calculation

Sources

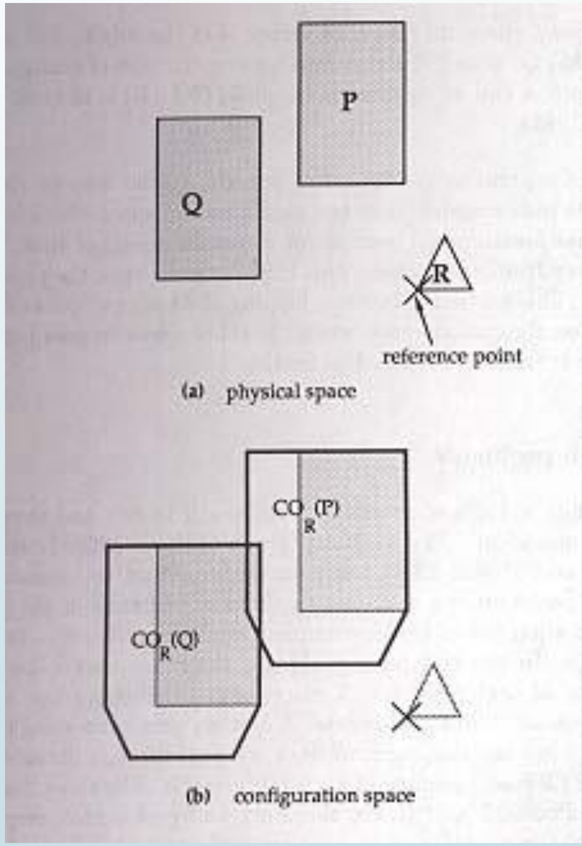
- 2.5. Task Space and Workspace – Modern Robotics (northwestern.edu)
- w3.cs.jmu.edu/spragunr/CS354_F22/readings/planning.pdf
- https://cs.stanford.edu/people/eroberts/courses/soco/projects/1998-99/robotics/definitions.html
- Configuration Space for Motion Planning, RSS Lecture 10, Prof. Seth Teller, MIT
- E190Q – Lecture 14 Autonomous Robot Navigation, Instructor: Chris Clark, Princeton University
- Robotic Motion Planning: Configuration Space Robotics Institute 16-735, Howie Choset, CMU
- A Modern Approach to Artificial Intelligence, Russell & Norvig
- Planning Algorithms, Steven M LaValle

Motion Planning in C Space

- Define C Space
- Define Start and Goal Configuration
- Define Obstacles
  - 1D case
  - 2D Space with assumption about shape of obstacles and robot
    - Star Algorithm
- Without defining obstacles
  - Sampling based strategy
  - Probe C Space to know distance from obstacles

The Configuration Space

- The configuration space is a transformation from the physical space in which the robot is of finite-size into another space in which the robot is treated as a point. In other words, the configuration space is obtained by shrinking the robot to a point, while growing the obstacles by the size of the robot.
- The figures illustrates the concept of configuration space. P and Q are fixed obstacles in physical space, and R is the robot, whose orientation is fixed.
- Figure b shows the corresponding configuration space.
- The free space of a configuration space simply consists of the areas not occupied by obstacles. Any configuration within this space is called a free configuration.
- The free path between an initial configuration and a goal configuration is the path which lies completely in free space and does not come into contact with any obstacles.



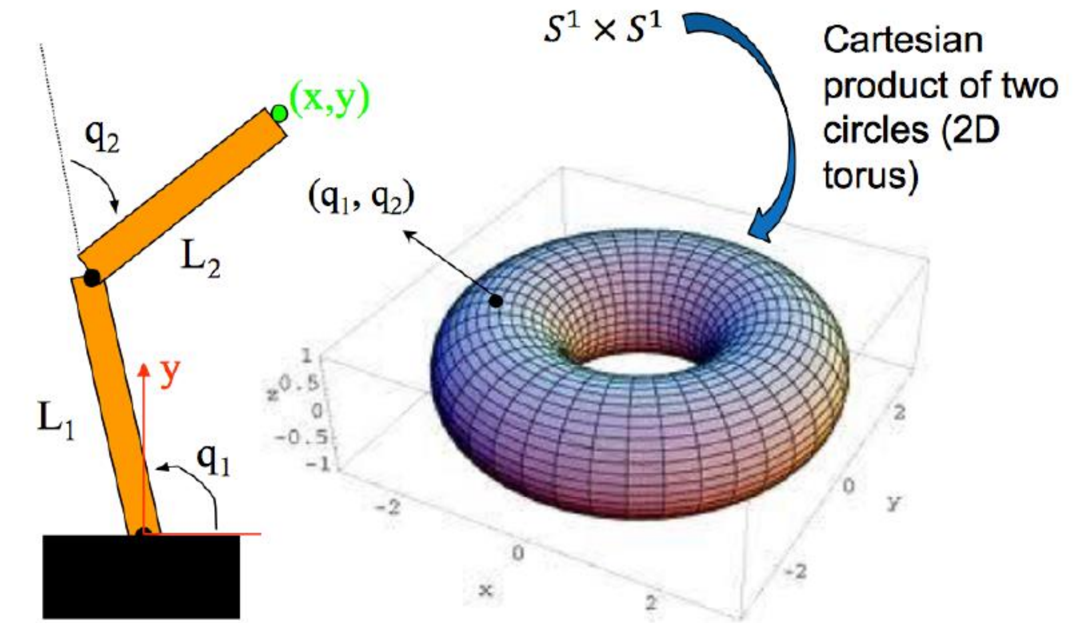
Task Space and Workspace

- The task space is a space in which the robot's task can be naturally expressed.
- For example, if the task is to control the position of the tip of a marker on a board, then task space is the Euclidean plane. If the task is to control the position and orientation of a rigid body, then the task space is the 6-dimensional space of rigid body configurations. One only has to know about the task, not the robot, to define the task space.
- The workspace is a specification of the configurations that the end-effector of the robot can reach, and has nothing to do with a particular task.
- The workspace is often defined in terms of the Cartesian points that can be reached by the end-effector, but it is also possible to include the orientation. The set of positions that can be reached with all possible orientations is sometimes called the dexterous workspace.

C-Space of Point Object & Robotic Arm

- The position of a single particle moving in ordinary Euclidean 3-space is defined by the vector q=(x,y,z)}, and therefore its configuration space is $Q = \mathbb{R}^3$
- If the particle is attached to a rigid linkage, free to swing about the origin, it is effectively constrained to lie on a sphere. Its configuration space is the subset of coordinates in $\mathbb{R}^3$ that define points on the sphere S^2
- In this case, one says that the manifold Q is the sphere, i.e. $Q = S^2$
- For a robotic arm consisting of numerous rigid linkages, the configuration space consists of the location of each linkage taken to be a rigid body, subject to the constraints of how the linkages are attached to each other, and their allowed range of motion. Thus, for n linkages, one might consider the total space $[\mathbb{R}^3 \times \mathrm{SO}(3)]^n$

Configuration Space & Motion Planning



Topology of Configuration Space

- To facilitate motion planning, the configuration space was defined as a tool that can be used with planning algorithms.
- A configuration *q* will completely define the state of a robot (e.g. mobile robot (*x*, *y*, *θ*)
- The configuration space C, is the space of all possible configurations of the robot.
- The free space $F \subseteq C$ is the portion of the free space which is collision-free.
- The goal of motion planning then, is to find a path in *F* that connects the initial configuration q_{start} to the goal configuration q_{goal}
- For a robot with *k* total motion DOFs, C-space is a coordinate system with *one dimension per DOF*

Obstacle Representation

Minkowski Sum

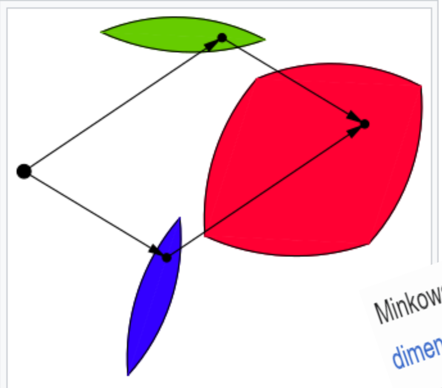
In geometry, the **Minkowski sum** of two sets of position vectors *A* and *B* in Euclidean space is formed by adding each vector in *A* to each vector in *B*:

$$A + B = \{\mathbf{a} + \mathbf{b} \,|\, \mathbf{a} \in A,\ \mathbf{b} \in B\}$$

The **Minkowski difference** (also *Minkowski subtraction*, *Minkowski decomposition*, or *geometric difference*)[1] is the corresponding inverse, where $(A - B)$ produces a set that could be summed with *B* to recover *A*. This is defined as the complement of the Minkowski sum of the complement of *A* with the reflection of *B* about the origin.[2]

$$-B = \{-\mathbf{b} \,|\, \mathbf{b} \in B\}$$
$$A - B = (A^{\complement} + (-B))^{\complement}$$

This definition allows a symmetrical relationship between the Minkowski sum and difference.



The red figure is the Minkowski sum of blue and green figures.

Minkowski is perhaps best known for his foundational work describing space and time as a four-dimensional space, now known as "Minkowski spacetime", which facilitated geometric interpretations of Albert Einstein's special theory of relativity (1905).

For example, if we have two sets *A* and *B*, each consisting of three position vectors (informally, three points), representing the vertices of two triangles in $\mathbb{R}^2$, with coordinates

$$A = \{(1,0), (0,1), (0,-1)\}$$

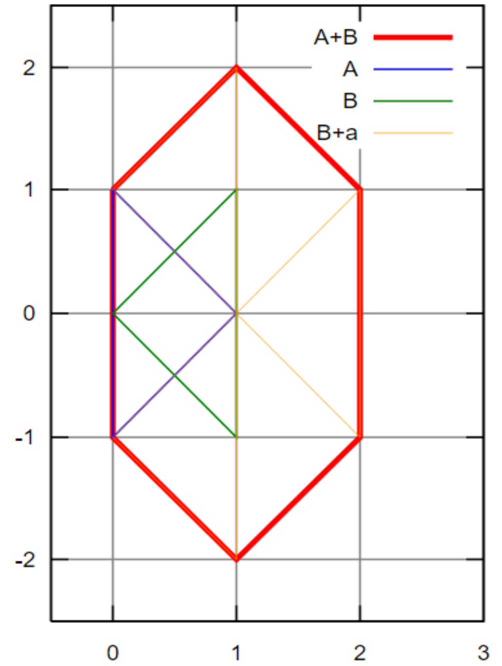
and

$$B = \{(0,0), (1,1), (1,-1)\}$$

then their Minkowski sum is

$$A + B = \{(1,0), (2,1), (2,-1), (0,1), (1,2), (1,0), (0,-1), (1,0), (1,-2)\},$$

which comprises the vertices of a hexagon and the three interior points of that hexagon having integral coordinates.



1D Case

- In Figure 4.12, both the robot A = [−1, 2] and obstacle region O = [0, 4] are intervals in a one-dimensional world, W = R. The negation, −A, of the robot is shown as the interval [−2, 1]. Finally, by applying the Minkowski sum to O and −A, the C_{space} obstacle, C_{obs} = [−2, 5], is obtained.

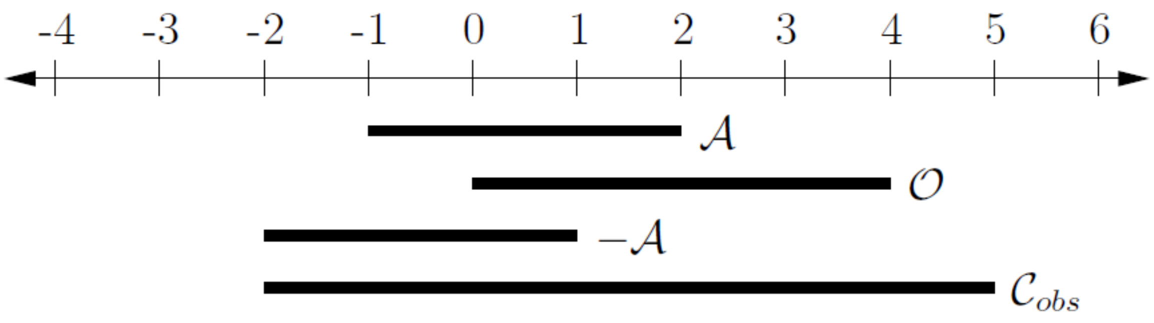
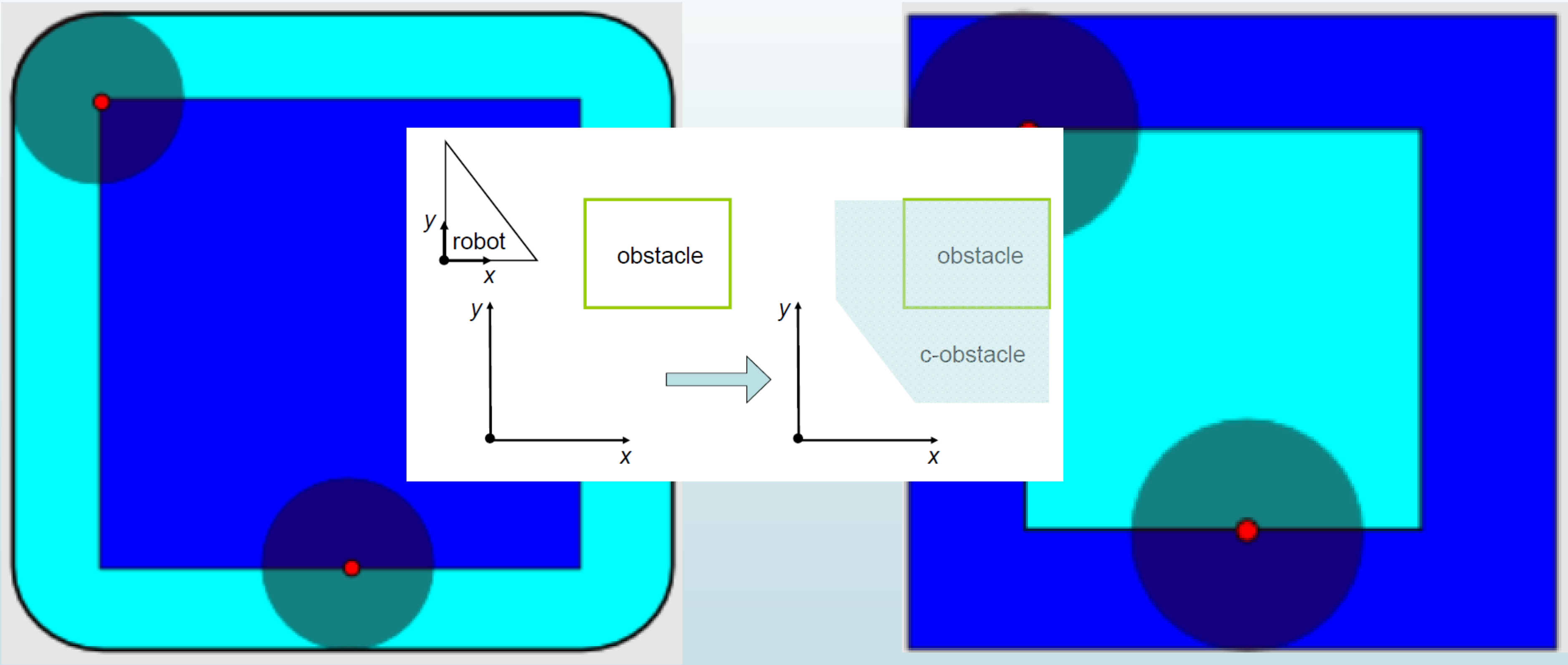


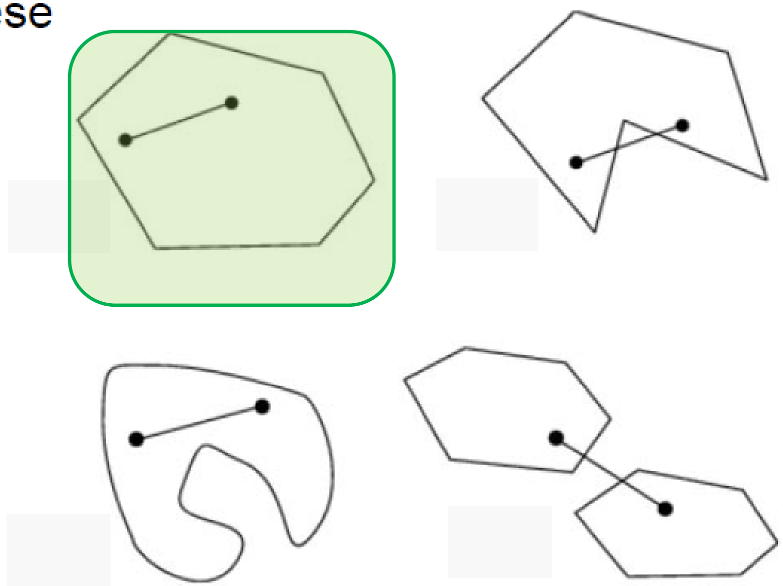
Figure 4.12: A one-dimensional C-space obstacle.

Revisiting Minkowski Sum in 2D
Dilation & Erosion



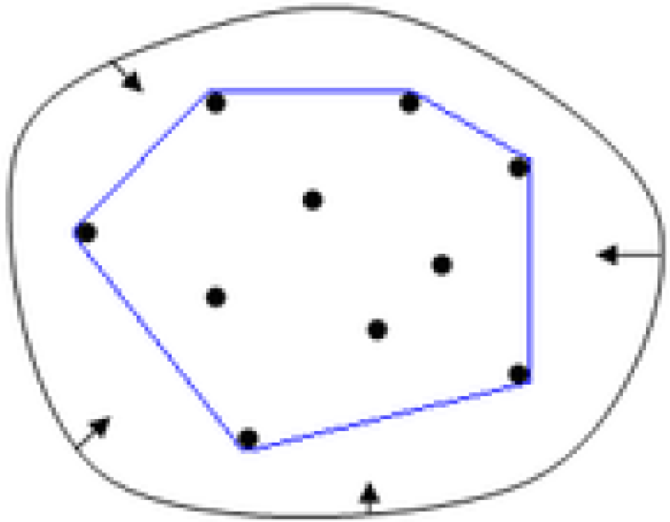
Convex Sets

- A set S is *convex* if and only if every line segment connecting two points in S is contained within S
- Which of these are convex?



Convex Hull of a Set of Points

- Intuition: shrink wrap or rubber band around points



Type of Robots

- **Holonomic constraints** result from physical restrictions that make it impossible for the robot to enter some regions of the configuration space.
- For example, the elbow joint on a robotic arm may only have a 90 degree range of motion. Holonomic constraints don't significantly complicate the path planning problem: we can simply extend our notion of C_{free} to exclude restricted regions.
- **Non-holonomic constraints** don't directly restrict which regions of the space are accessible. Instead, they restrict how the robot can move from one configuration to another.
- A classic example of a non-holonomic constraint is the inability of a car to slide sideways into a parking spot. There is no constraint preventing the car from being in the parking spot, but the mechanics of the vehicle prevent it from following a straight-line path to the desired configuration.
- Non-holonomic constraints can also arise from system dynamics: A vehicle moving at 5 miles per hour can easily make a 30 degree turn, while a vehicle moving at 50 miles per hour would roll over.
- Non-holonomic constraints of this sort are referred to as kino-dynamic constraints. Non-holonomic constraints complicate the path planning problem and require the use of specialized algorithms.

2D Case

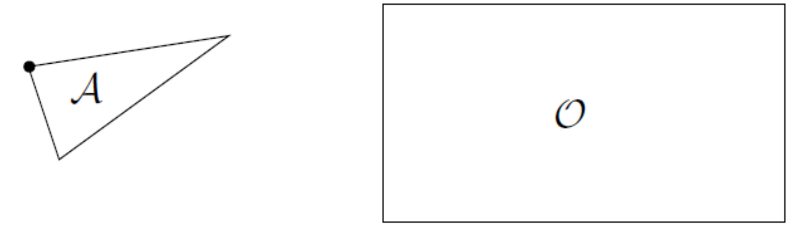


Figure 4.13: A triangular robot and a rectangular obstacle.

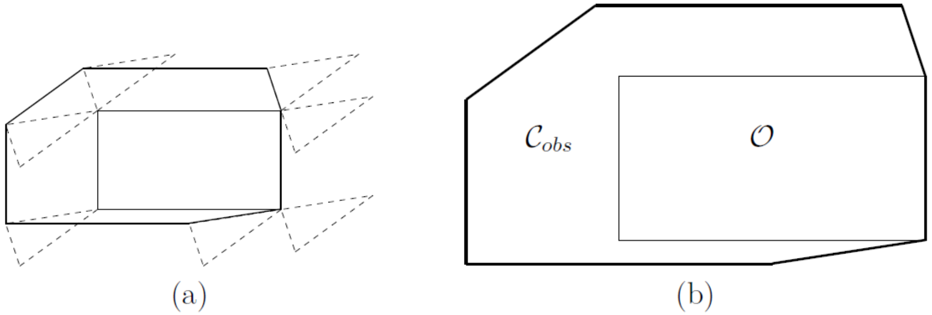
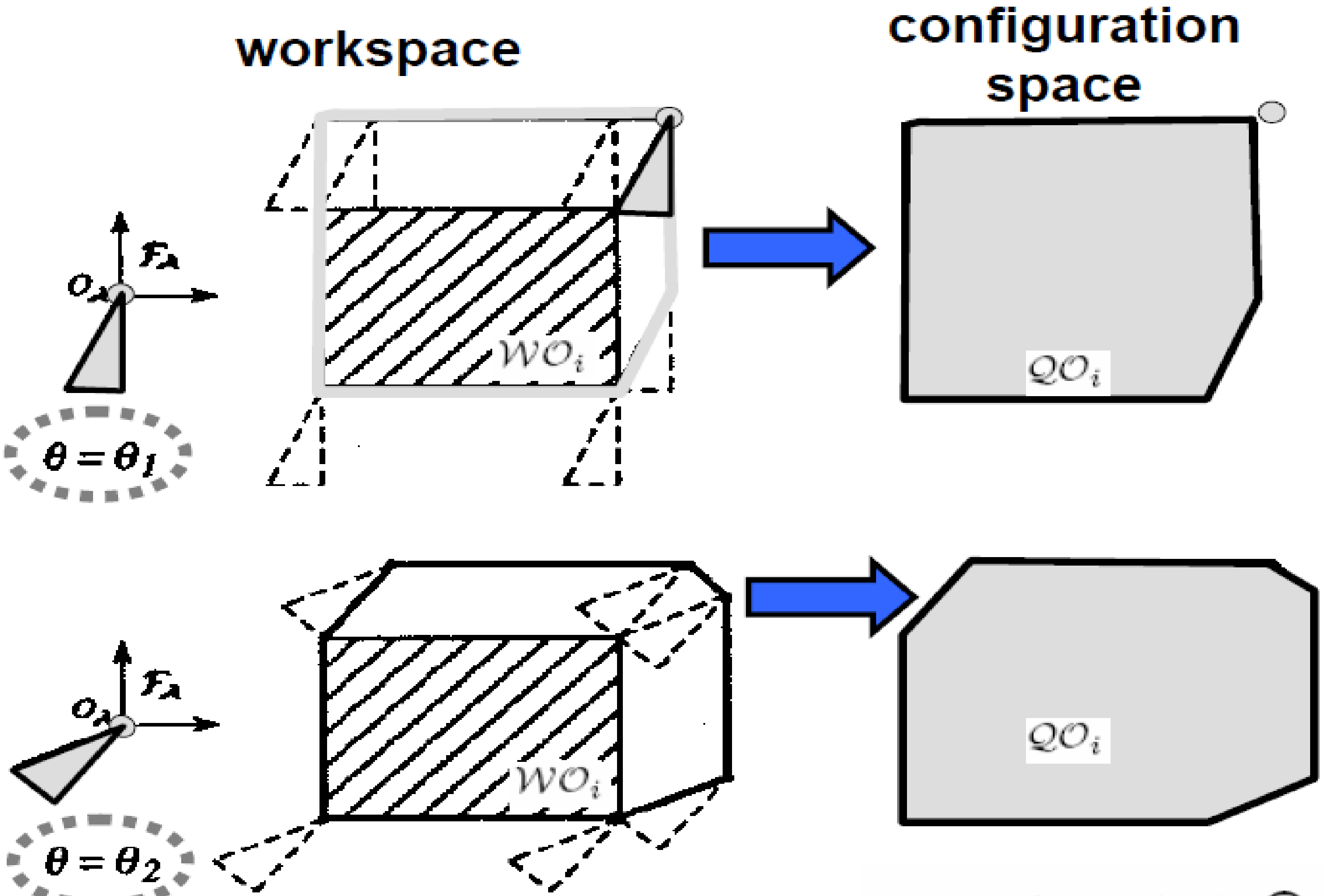


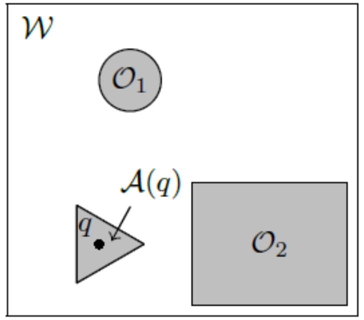
Figure 4.14: (a) Slide the robot around the obstacle while keeping them both in contact. (b) The edges traced out by the origin of $\mathcal{A}$ form $\mathcal{C}_{obs}$.

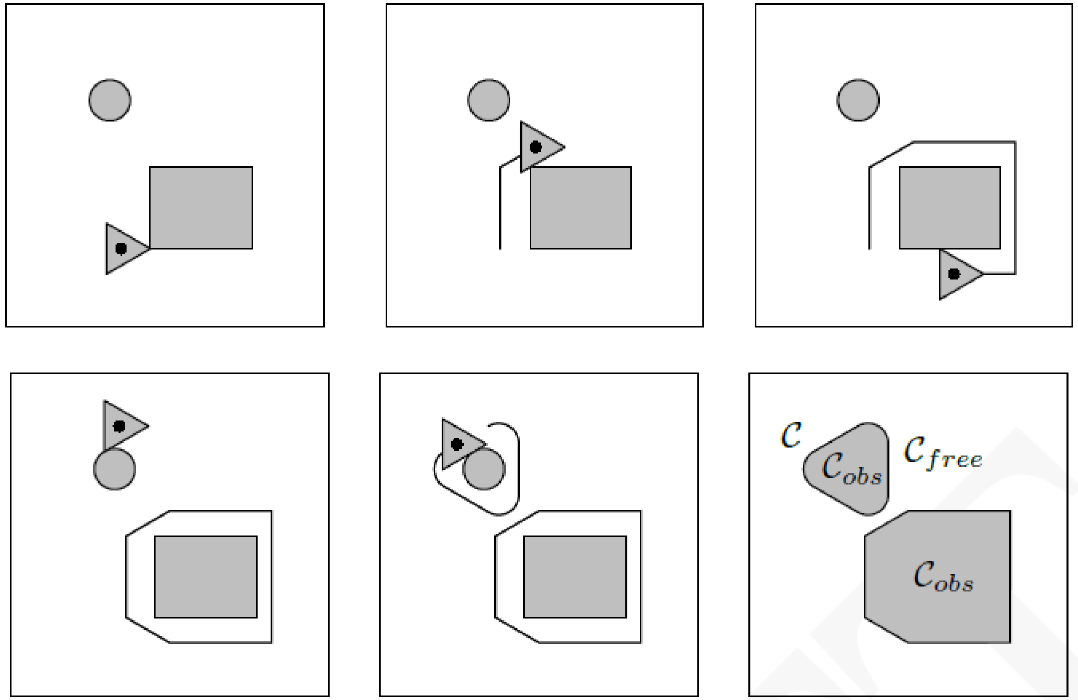
- Convex polygonal obstacle O
- Convex polygonal robot A
- Nonconvex obstacles and robots can be modeled as the union of convex parts.
- C_{obs} can be considered as the union of convex components, each of which corresponds to a convex component of A colliding with a convex component of O

Polygonal robot translating & rotating in 2-D workspace



$$\mathcal{QO}_i = \{q \in \mathcal{Q} \mid R(q) \bigcap \mathcal{WO}_i \neq \emptyset\}.$$





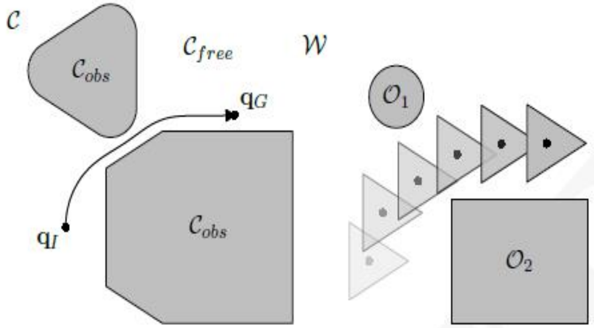
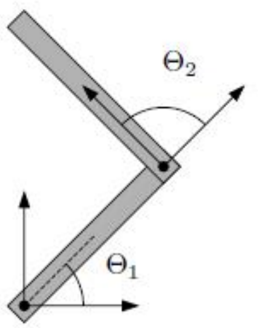
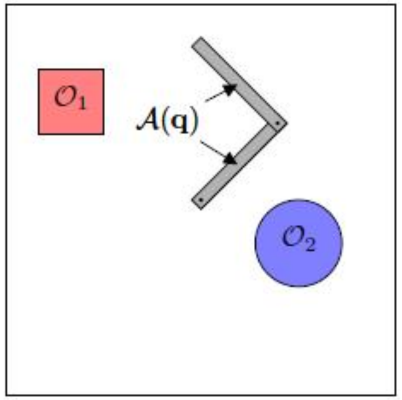


Figure 4.4: Left: A valid path from an initial configuration $\mathbf{q}_I$ to a goal configuration $\mathbf{q}_G$. Right: The robot trajectory corresponding to the indicated path.

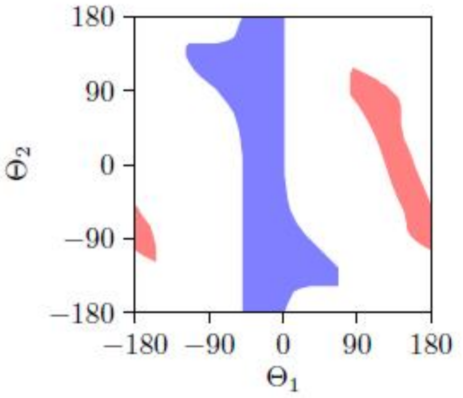
Figure 4.2: Determining $\mathcal{C}_{obs}$. The bottom-right figure illustrates $\mathcal{C}_{obs}$ and $\mathcal{C}_{free}$ for the environment shown in Figure 4.1.



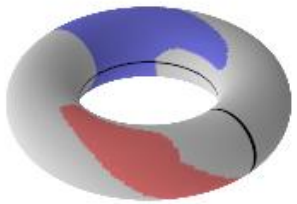
(a) 2D robot arm with two degrees of freedom. For this robot $\mathbf{q} = [\Theta_1, \Theta_2]^T$.



(b) An example of a possible configuration of the arm along with a pair of obstacles. Notice that $\mathcal{O}_2$ is close enough to the arm to prevent the first link one from making a full rotation.



(c) An illustration of $\mathcal{C}_{obs}$ for the robot configuration illustrated in 4.3b. The colors indicate configurations that intersect with the corresponding objects in 4.3b.



(d) The configuration space from 4.3c represented as a torus. The black lines are located at $\theta_1 = 180/-180$ and $\theta_2 = 180/-180$. We could imagine creating 4.3c by cutting along these lines and unwrapping the surface.

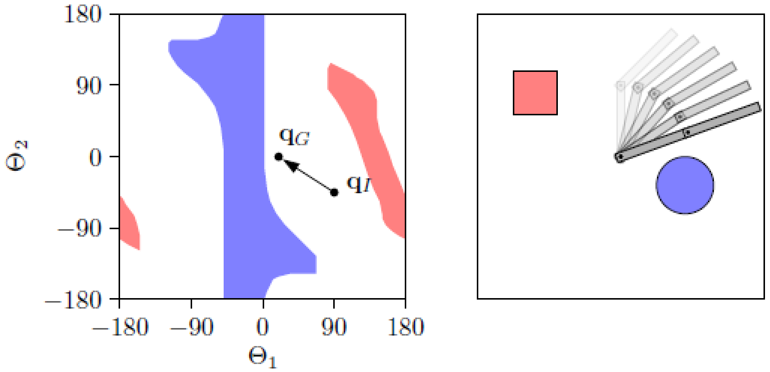


Figure 4.5: Left: A valid path from an initial configuration $\mathbf{q}_I$ to a goal configuration $\mathbf{q}_G$. Right: The robot trajectory corresponding to the indicated path.

Star Algorithm

- Every edge from O and A is used exactly once in the construction of C_{obs}
- Let $\alpha_1, \alpha_2, \ldots, \alpha_n$ denote the angles of the inward edge normal in counterclockwise order around A.
- Let $\beta_1, \beta_2, \ldots, \beta_n$ denote the outward edge normals to O.
- After sorting both sets of angles in circular order around S^1, C_{obs} can be constructed incrementally by using the edges that correspond to the sorted normals, in the order in which they are encountered.

Star Algorithm

The method is based on sorting normals to the edges of the polygons on the basis of angles.

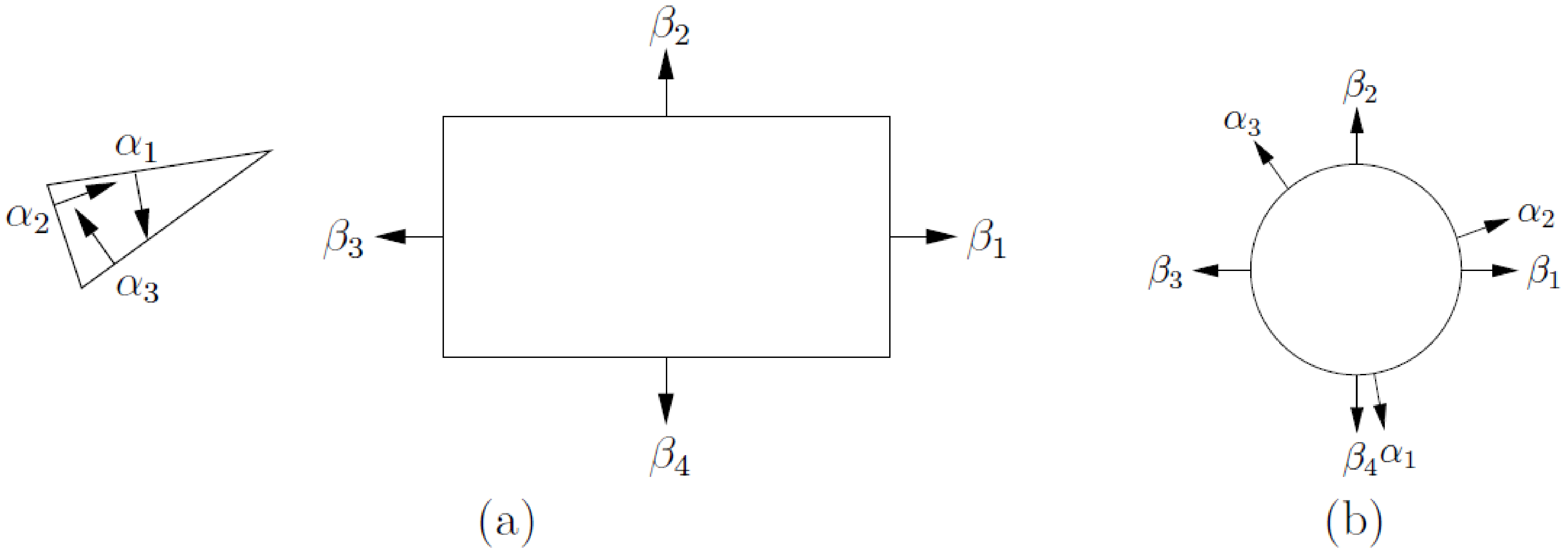


Figure 4.15: (a) Take the inward edge normals of $\mathcal{A}$ and the outward edge normals of $\mathcal{O}$. (b) Sort the edge normals around $\mathbb{S}^1$. This gives the order of edges in $\mathcal{C}_{obs}$.

Obstacle Contour

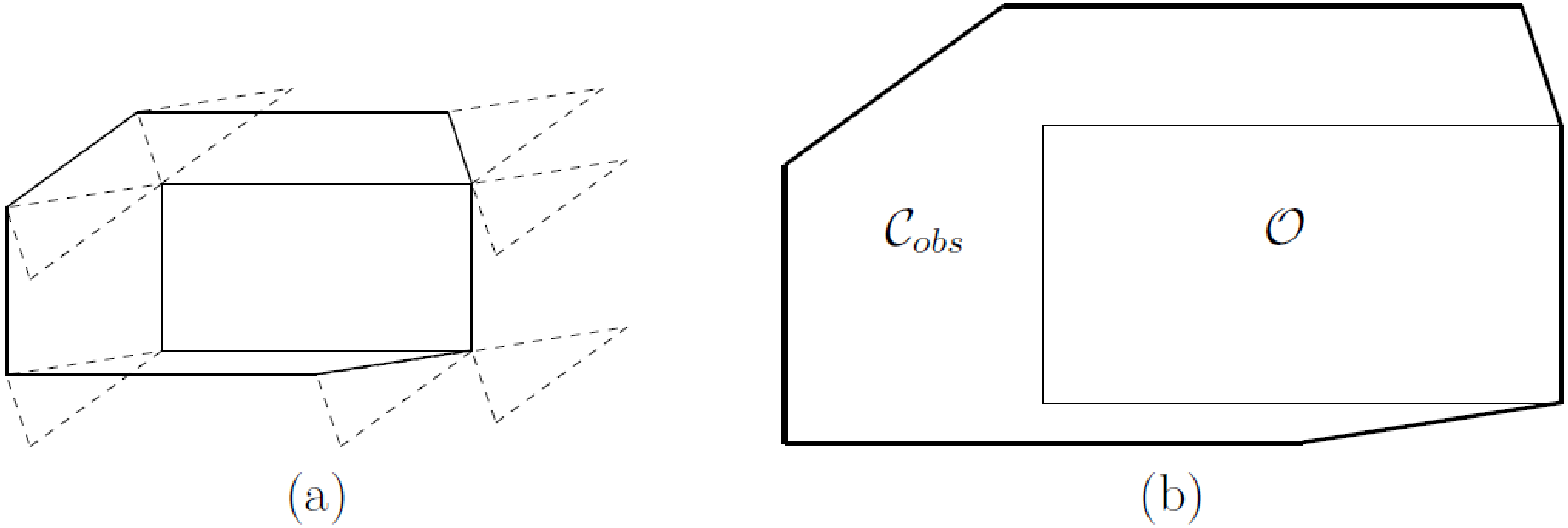


Figure 4.14: (a) Slide the robot around the obstacle while keeping them both in contact. (b) The edges traced out by the origin of $\mathcal{A}$ form $\mathcal{C}_{obs}$.

Types of Contacts

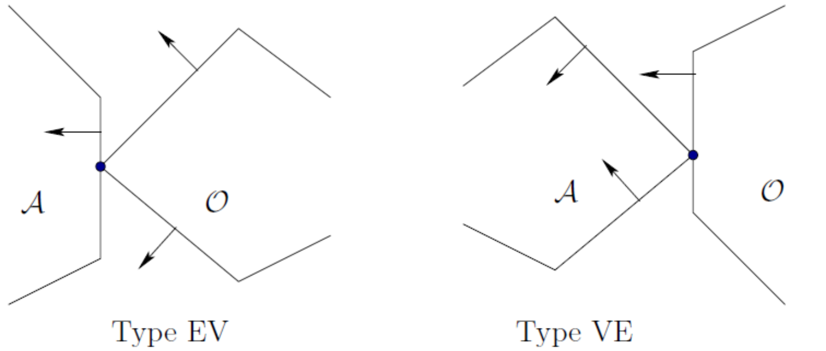


Figure 4.16: Two different types of contact, each of which generates a different kind of $\mathcal{C}_{obs}$ edge [280, 657].

- Type EV contact refers to the case in which an edge of A is in contact with a vertex of O.
- Type EV contacts contribute to n edges of Cobs, once for each edge of A.
- Type VE contact refers to the case in which a vertex of A is in contact with an edge of O. This contributes to m edges of Cobs.
- The basic principle is that the motion vector of the robot, as it slides around the obstacle, should be perpendicular to the vector normal to the surface of the obstacle in the VE case.
- In the EV case, the motion vector (considered at point p) should be perpendicular to the vector normal to the surface of the robot.

Contour Calculation

(without rotation)

- The normal vector n does not depend on the configuration of A because the robot cannot rotate.
- The vector v, however, depends on the translation $q = (x_t, y_t)$ of the point p.
- if the coordinates of p are (1, 2) for A(0, 0), then the expression for p at configuration (x_t, y_t) is $(1 + x_t, 2 + y_t)$.
- The obstacle region Cobs can be completely characterized by intersecting the resulting half-planes for each of the Type EV and Type VE contacts.
- This yields a convex polygon in C that has n + m sides

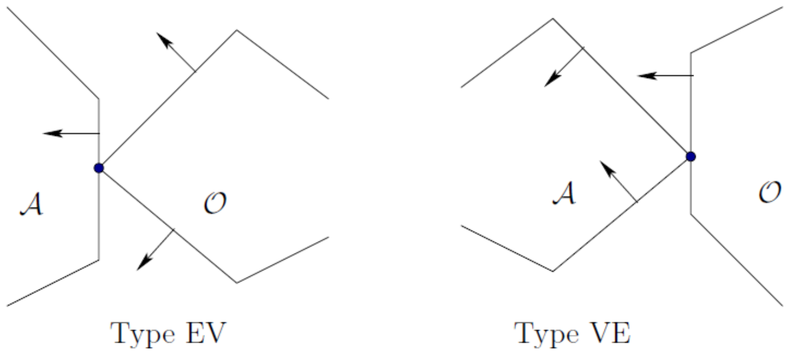


Figure 4.16: Two different types of contact, each of which generates a different kind of $\mathcal{C}_{obs}$ edge [280, 657].

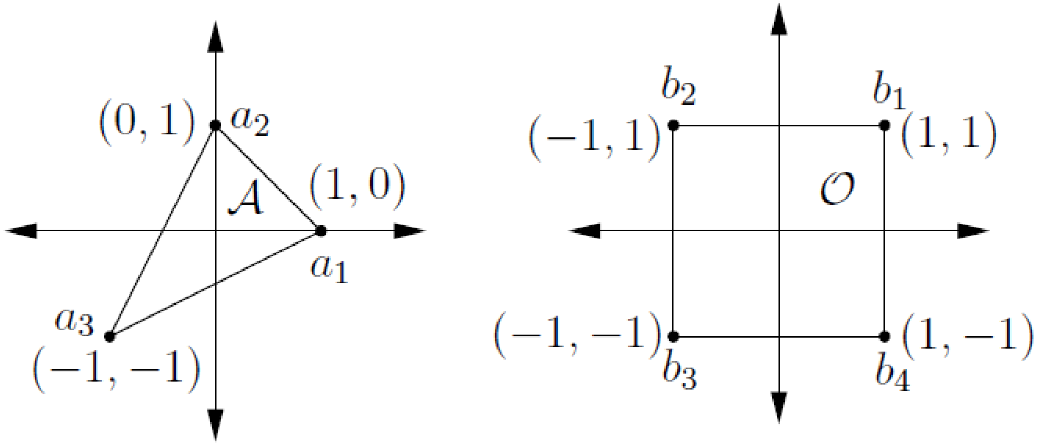
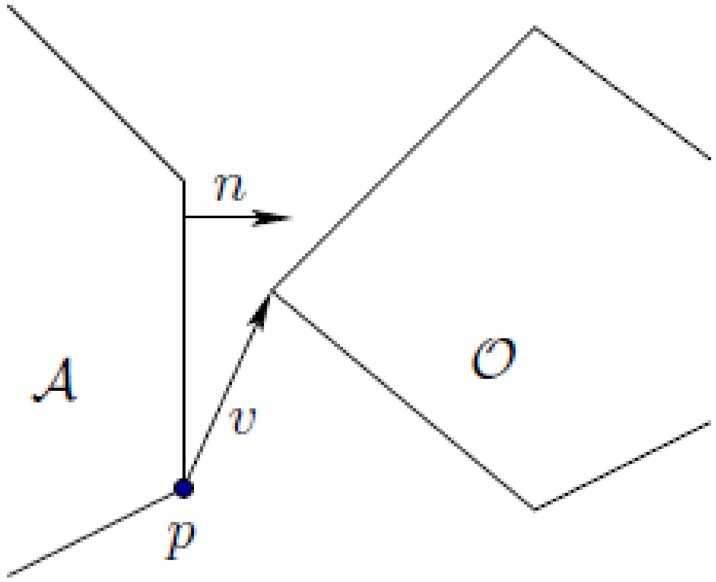
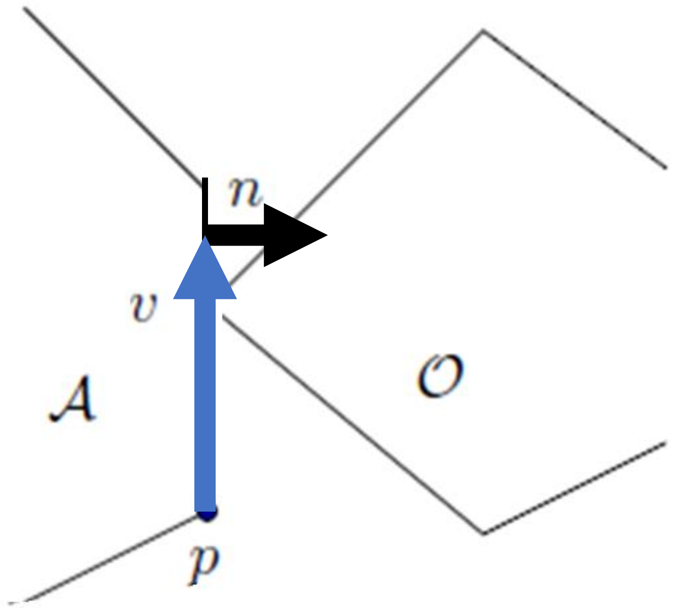


Figure 4.18: Consider constructing the obstacle region for this example.

Vector v and Normal n





Type	Vtx.	Edge	n	v	Half-Plane
VE	a_3	b_4-b_1	$[1, 0]$	$[x_t - 2, y_t]$	$\{q \in \mathcal{C} \mid x_t - 2 \leq 0\}$
VE	a_3	b_1-b_2	$[0, 1]$	$[x_t - 2, y_t - 2]$	$\{q \in \mathcal{C} \mid y_t - 2 \leq 0\}$
EV	b_2	a_3-a_1	[1,-2]	$[-x_t, 2 - y_t]$	$\{q \in \mathcal{C} \mid -x_t + 2y_t - 4 \leq 0\}$

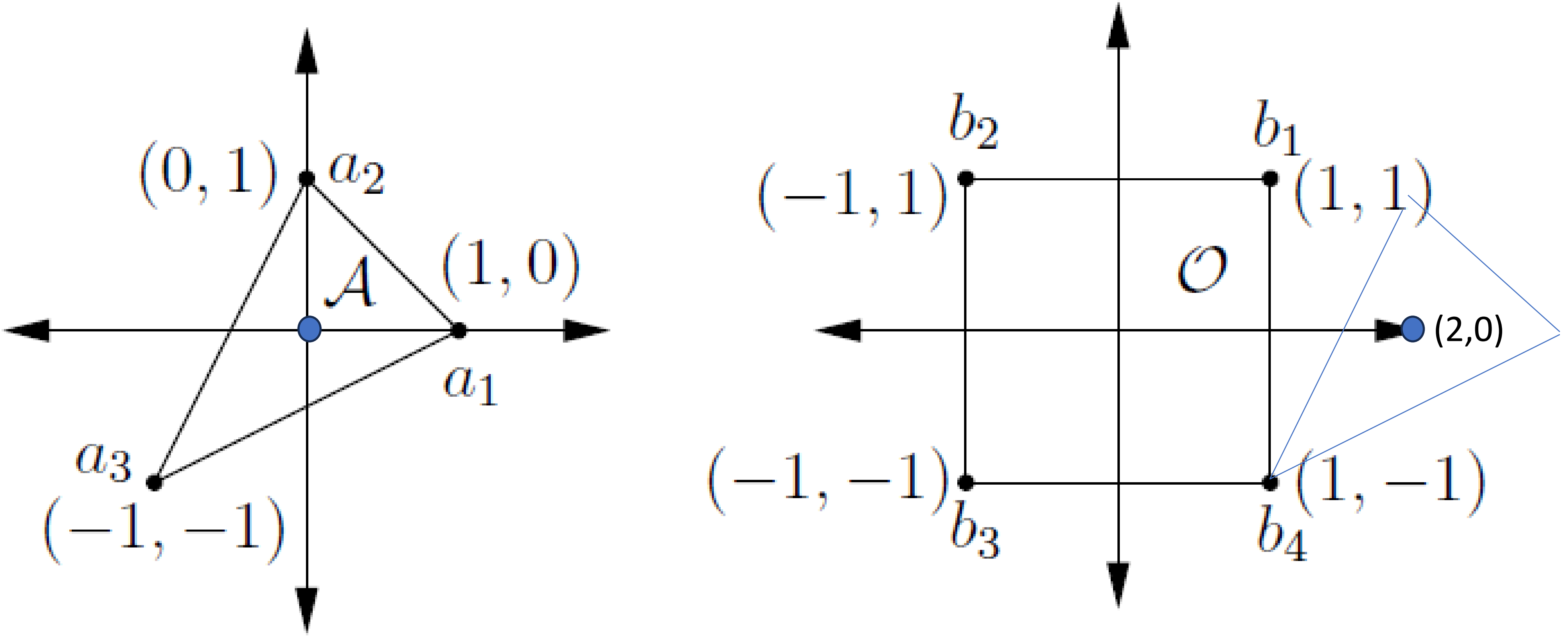


Figure 4.18: Consider constructing the obstacle region for this example.

Type	Vtx.	Edge	n	v	Half-Plane
VE	a_3	b_4-b_1	$[1, 0]$	$[x_t - 2, y_t]$	$\{q \in \mathcal{C} \mid x_t - 2 \leq 0\}$
VE	a_3	b_1-b_2	$[0, 1]$	$[x_t - 2, y_t - 2]$	$\{q \in \mathcal{C} \mid y_t - 2 \leq 0\}$
EV	b_2	a_3-a_1	[1,-2]	$[-x_t, 2 - y_t]$	$\{q \in \mathcal{C} \mid -x_t + 2y_t - 4 \leq 0\}$

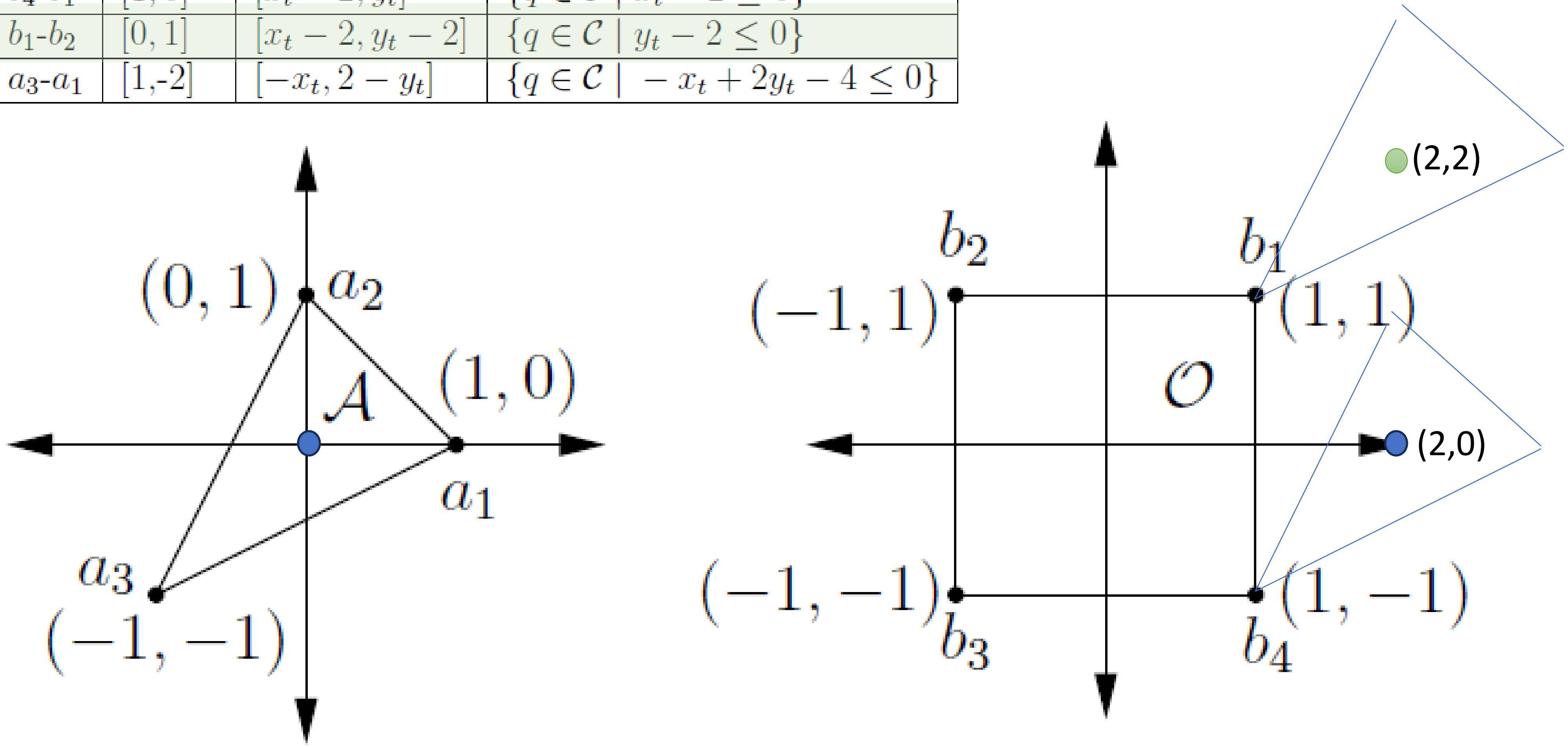


Figure 4.18: Consider constructing the obstacle region for this example.

Type	Vtx.	Edge	n	v	Half-Plane
VE	a_3	b_4-b_1	$[1, 0]$	$[x_t - 2, y_t]$	$\{q \in \mathcal{C} \mid x_t - 2 \leq 0\}$
VE	a_3	b_1-b_2	$[0, 1]$	$[x_t - 2, y_t - 2]$	$\{q \in \mathcal{C} \mid y_t - 2 \leq 0\}$
EV	b_2	a_3-a_1	$[1,-2]$	$[-x_t, 2 - y_t]$	$\{q \in \mathcal{C} \mid -x_t + 2y_t - 4 \leq 0\}$

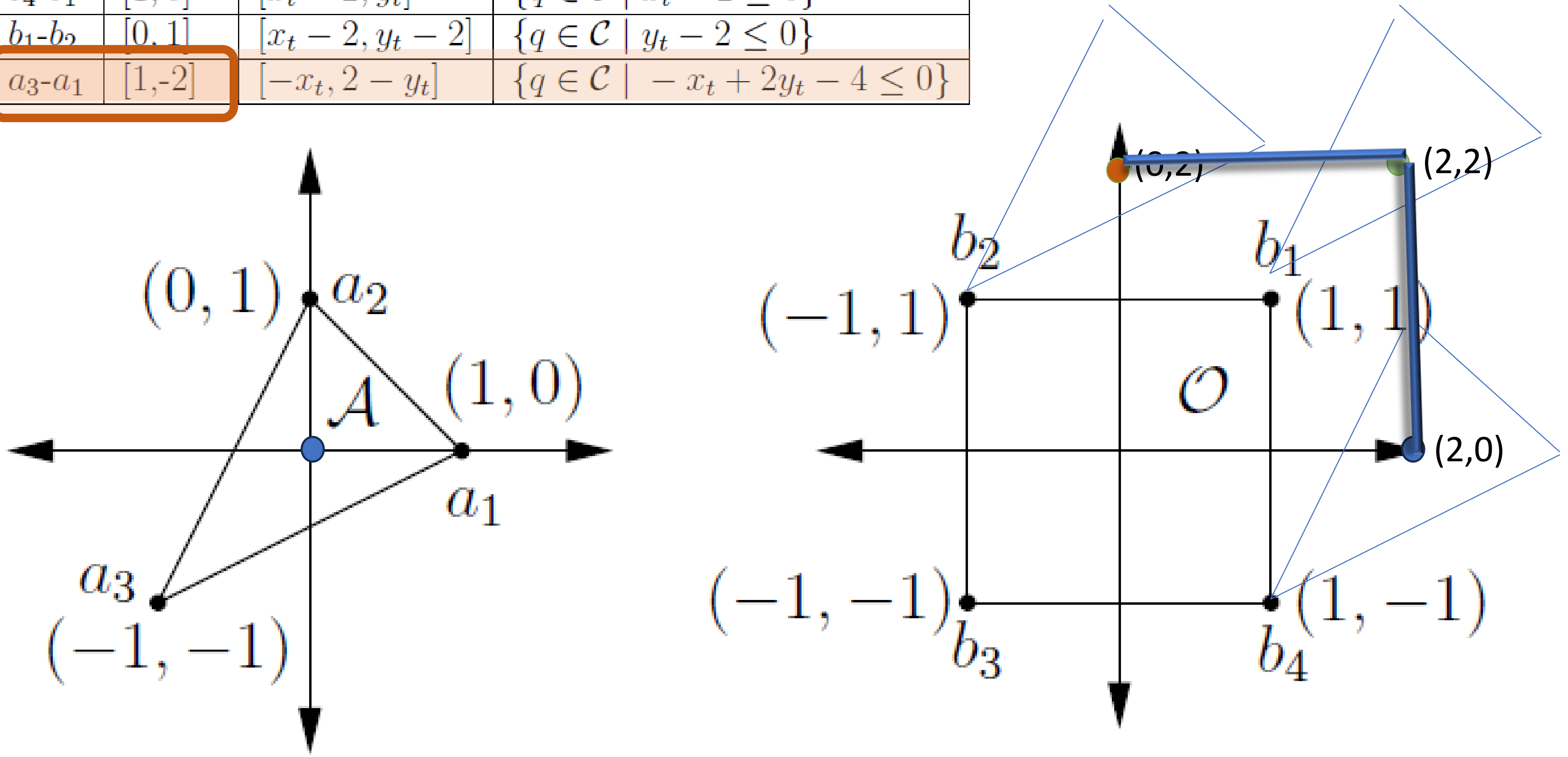
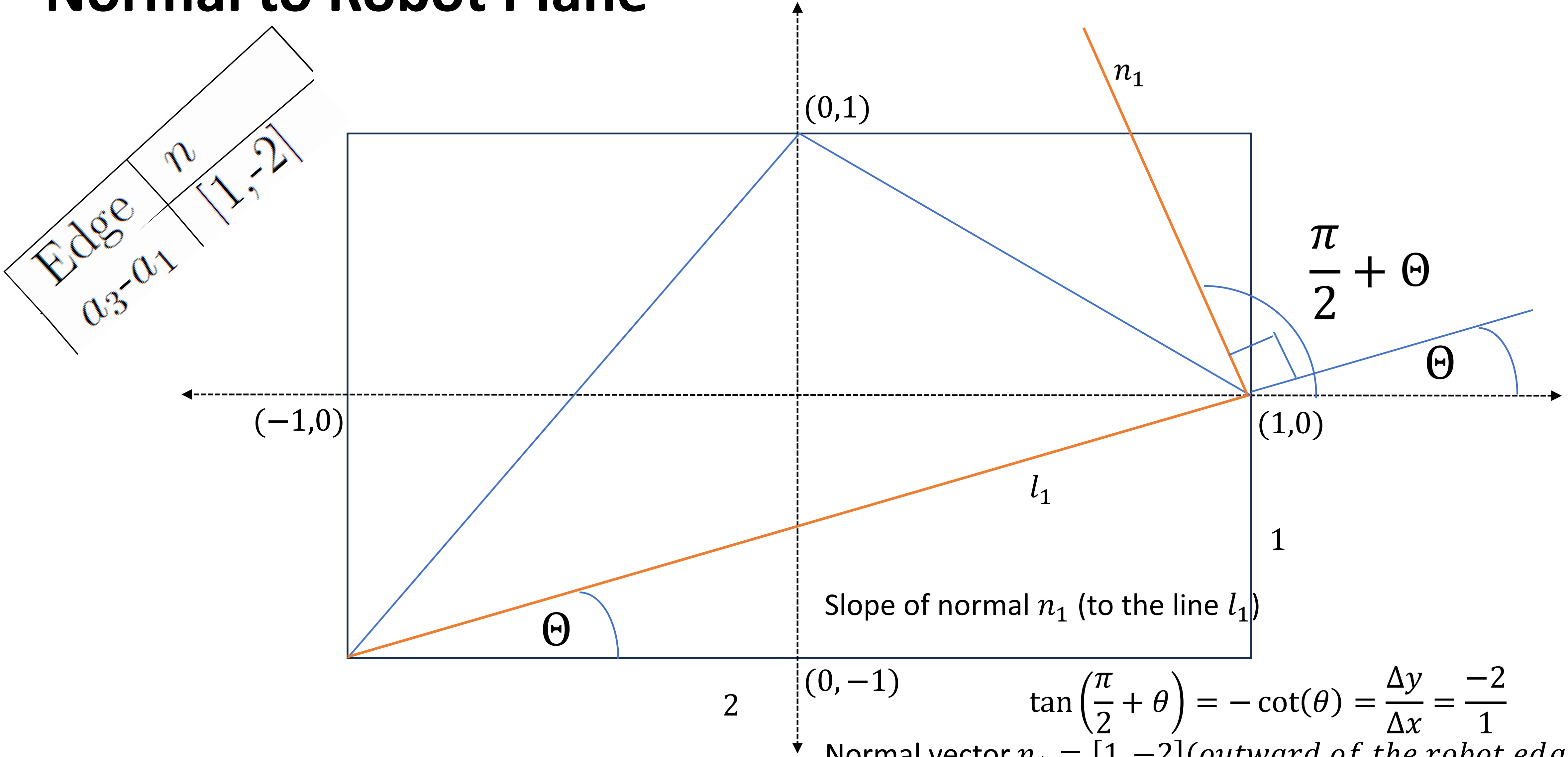


Figure 4.18: Consider constructing the obstacle region for this example.

Normal to Robot Plane



Slope of normal n_1 (to the line l_1)

$$\tan\left(\frac{\pi}{2}+\theta\right) = -\cot(\theta) = \frac{\Delta y}{\Delta x} = \frac{-2}{1}$$

Normal vector $n_1 = [1, -2]$ $(outward\ of\ the\ robot\ edge)$

VE	a_3	b_4-b_1	$[1, 0]$	$[x_t - 2, y_t]$	$\{q \in \mathcal{C} \mid x_t - 2 \leq 0\}$
VE	a_3	b_1-b_2	$[0, 1]$	$[x_t - 2, y_t - 2]$	$\{q \in \mathcal{C} \mid y_t - 2 \leq 0\}$
EV	b_2	a_3-a_1	$[1,-2]$	$[-x_t, 2 - y_t]$	$\{q \in \mathcal{C} \mid -x_t + 2y_t - 4 \leq 0\}$
VE	a_1	b_2-b_3	$[-1, 0]$	$[2 + x_t, y_t - 1]$	$\{q \in \mathcal{C} \mid -x_t - 2 \leq 0\}$

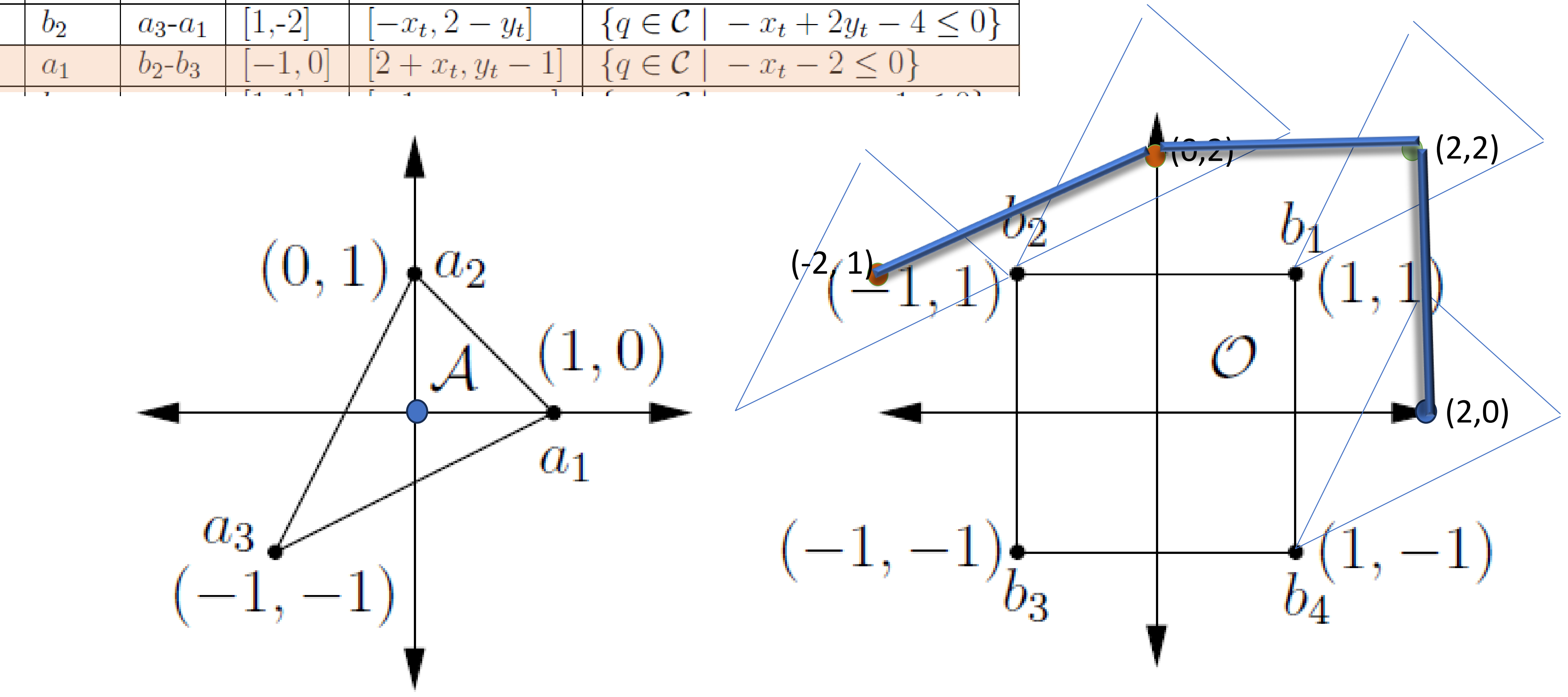


Figure 4.18: Consider constructing the obstacle region for this example.

VE	a_1	b_2-b_3	$[-1,0]$	$[2+x_t, y_t-1]$	$\{q \in \mathcal{C} \mid -x_t-2 \leq 0\}$
EV	b_3	a_1-a_2	$[1,1]$	$[-1-x_t, -y_t]$	$\{q \in \mathcal{C} \mid -x_t-y_t-1 \leq 0\}$
VE	a_2	b_3-b_4	$[0,-1]$	$[x_t+1, y_t+2]$	$\{q \in \mathcal{C} \mid -y_t-2 \leq 0\}$
EV	b_4	a_2-a_3	$[-2,1]$	$[2-x_t, -y_t]$	$\{q \in \mathcal{C} \mid 2x_t-y_t-4 \leq 0\}$

$-x_t - y_t - 3 \leq 0$

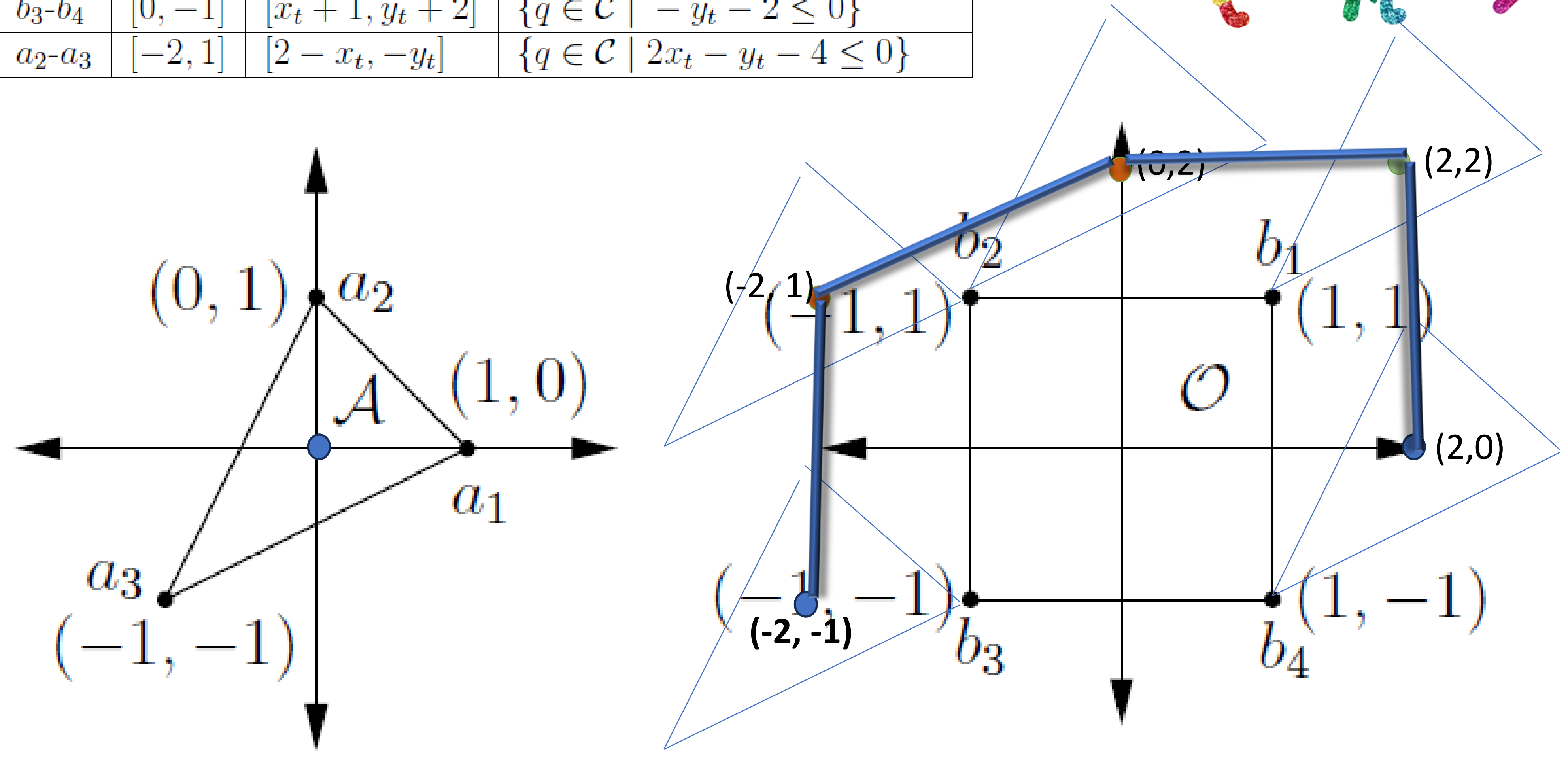


Figure 4.18: Consider constructing the obstacle region for this example.

VE	a_1	b_2-b_3	$[-1, 0]$	$[2 + x_t, y_t - 1]$	$\{q \in \mathcal{C} \mid -x_t - 2 \leq 0\}$
EV	b_3	a_1-a_2	$[1, 1]$	$[-1 - x_t, -y_t]$	$\{q \in \mathcal{C} \mid -x_t - y_t - 1 \leq 0\}$
VE	a_2	b_3-b_4	$[0, -1]$	$[x_t + 1, y_t + 2]$	$\{q \in \mathcal{C} \mid -y_t - 2 \leq 0\}$
EV	b_4	a_2-a_3	$[-2, 1]$	$[2 - x_t, -y_t]$	$\{q \in \mathcal{C} \mid 2x_t - y_t - 4 \leq 0\}$

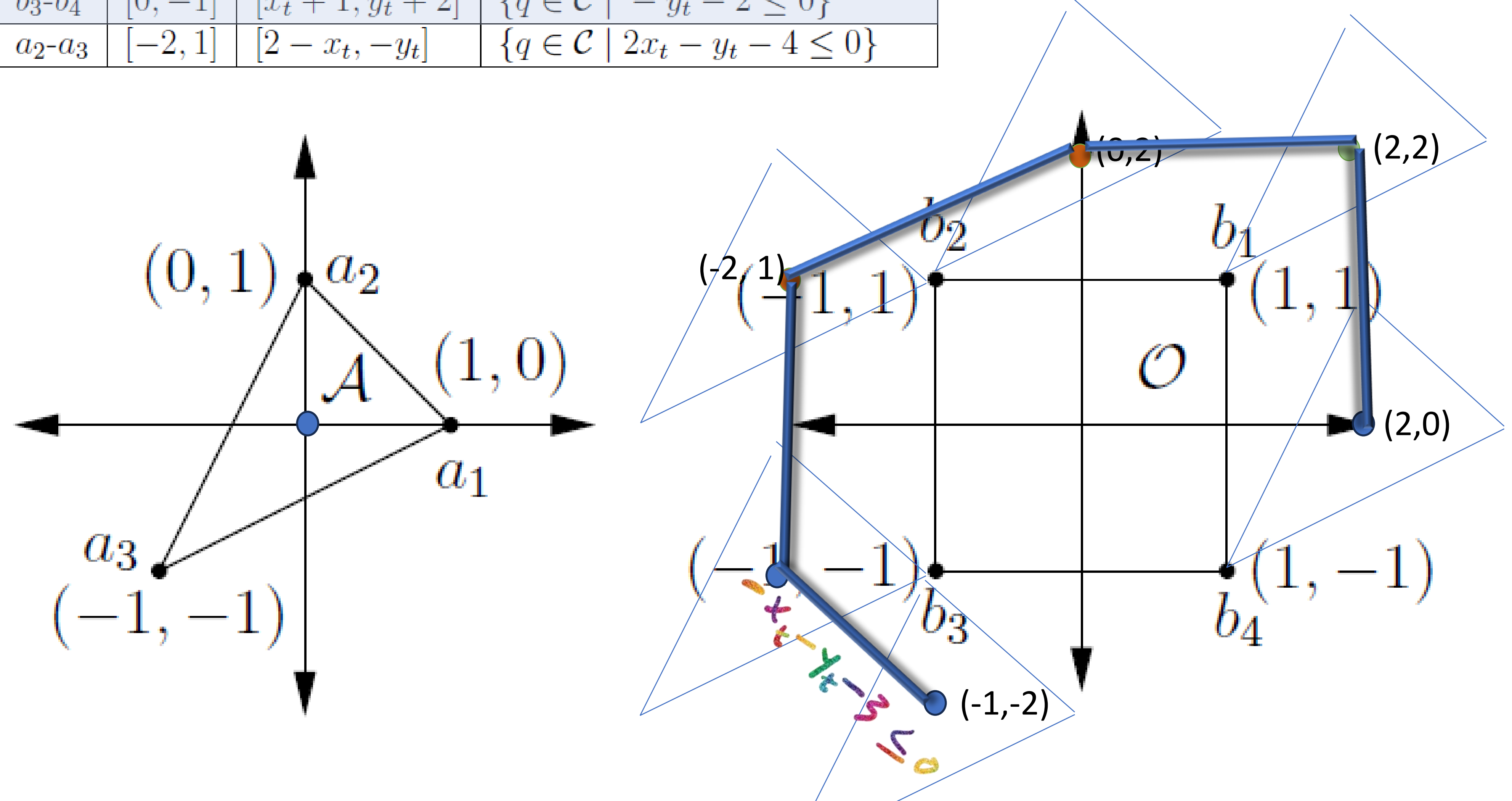


Figure 4.18: Consider constructing the obstacle region for this example.

VE	a_1	b_2-b_3	$[-1, 0]$	$[2 + x_t, y_t - 1]$	$\{q \in \mathcal{C} \mid -x_t - 2 \leq 0\}$
EV	b_3	a_1-a_2	$[1, 1]$	$[-1 - x_t, -y_t]$	$\{q \in \mathcal{C} \mid -x_t - y_t - 1 \leq 0\}$
VE	a_2	b_3-b_4	$[0, -1]$	$[x_t + 1, y_t + 2]$	$\{q \in \mathcal{C} \mid -y_t - 2 \leq 0\}$
EV	b_4	a_2-a_3	$[-2, 1]$	$[2 - x_t, -y_t]$	$\{q \in \mathcal{C} \mid 2x_t - y_t - 4 \leq 0\}$

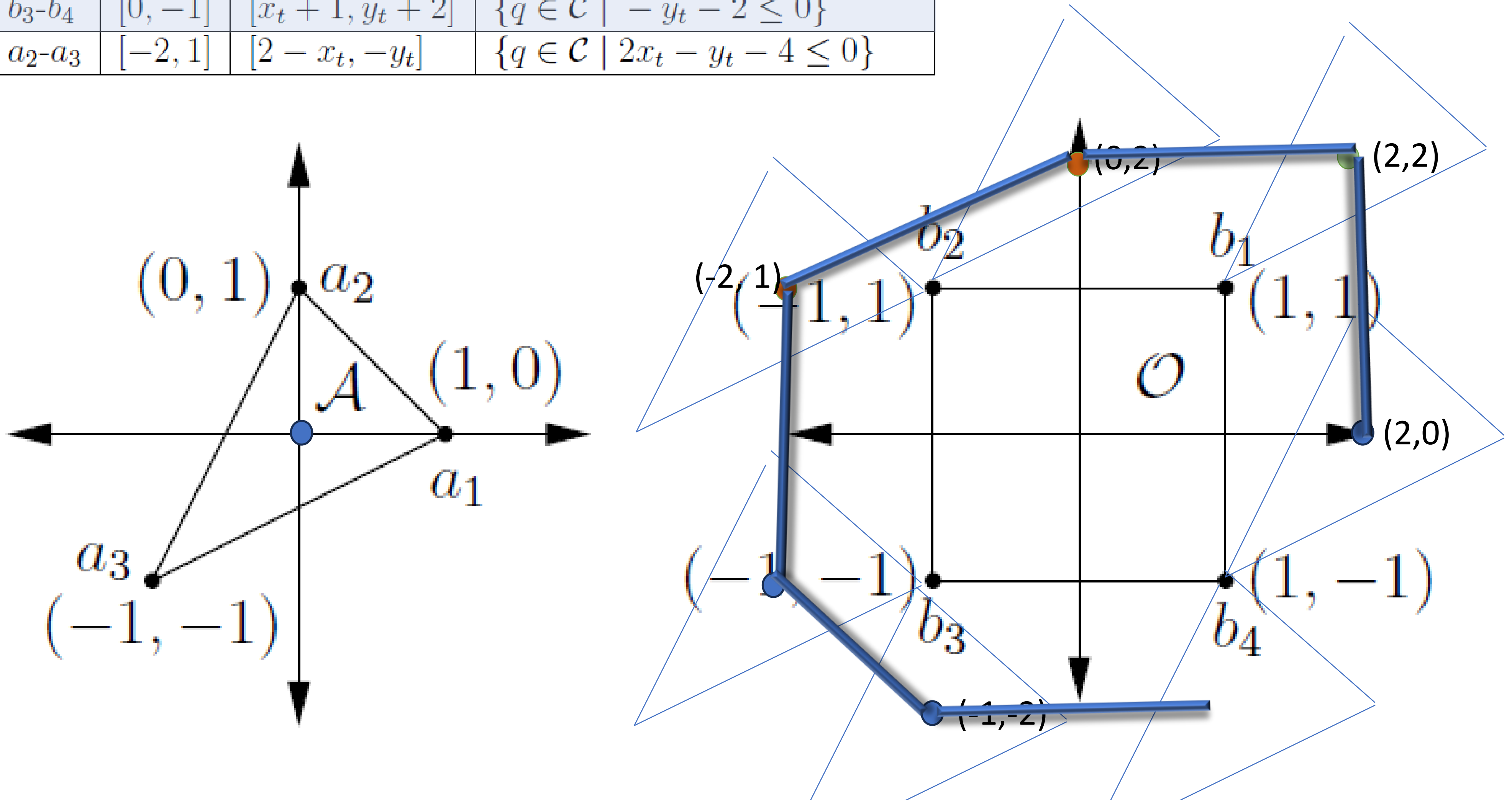


Figure 4.18: Consider constructing the obstacle region for this example.

VE	a_1	b_2-b_3	$[-1, 0]$	$[2 + x_t, y_t - 1]$	$\{q \in \mathcal{C} \mid -x_t - 2 \leq 0\}$
EV	b_3	a_1-a_2	$[1, 1]$	$[-1 - x_t, -y_t]$	$\{q \in \mathcal{C} \mid -x_t - y_t - 1 \leq 0\}$
VE	a_2	b_3-b_4	$[0, -1]$	$[x_t + 1, y_t + 2]$	$\{q \in \mathcal{C} \mid -y_t - 2 \leq 0\}$
EV	b_4	a_2-a_3	$[-2, 1]$	$[2 - x_t, -y_t]$	$\{q \in \mathcal{C} \mid 2x_t - y_t - 4 \leq 0\}$

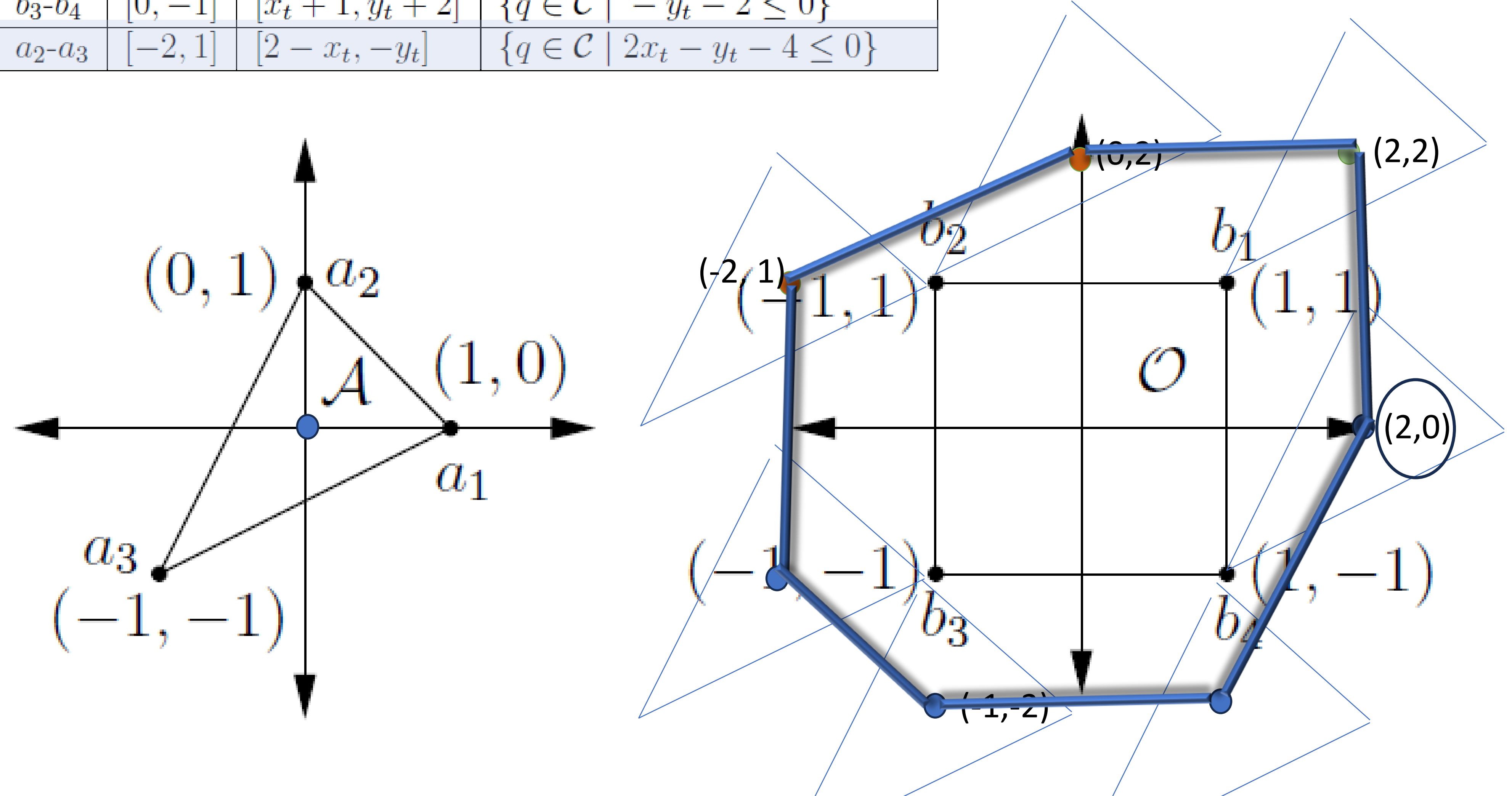


Figure 4.18: Consider constructing the obstacle region for this example.

Polyhedral Obstacle

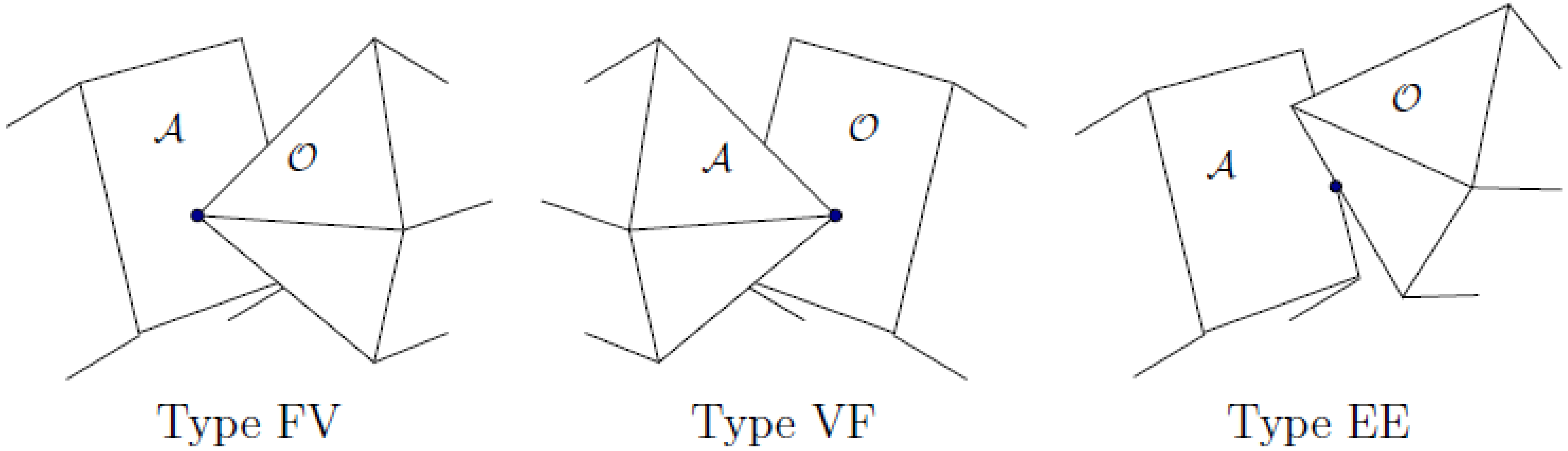
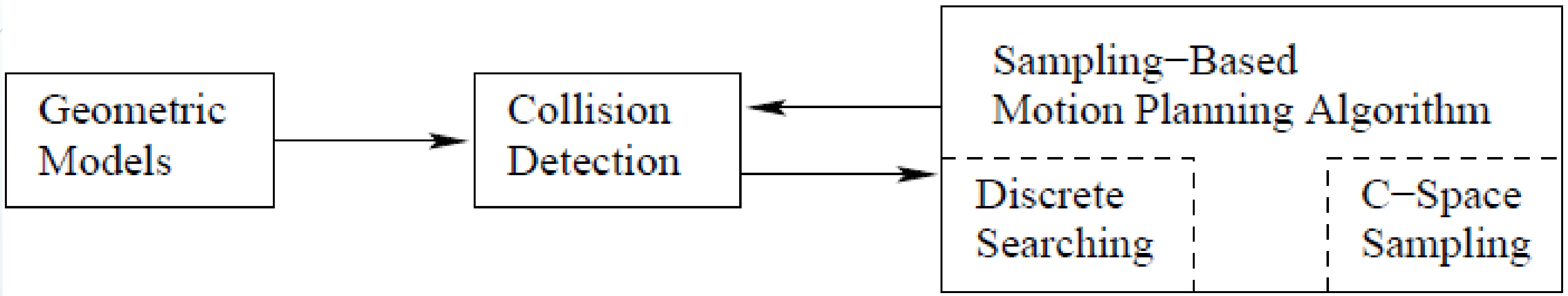


Figure 4.20: Three different types of contact, each of which generates a different kind of $\mathcal{C}_{obs}$ face.

1. Type FV: A face of A and a vertex of O
2. Type VF: A vertex of A and a face of O
3. Type EE: An edge of A and an edge of O

Sampling Based Strategy



- Avoid the explicit construction of C_{obs}
- Conduct a search that probes the C-space with a sampling scheme
- A collision detection module handles concerns such as whether the models are semi-algebraic sets, 3D triangles, nonconvex polyhedra, and so on.
- The motion planning algorithm considers the collision detection module as a "black box." This enables the development of planning algorithms that are independent of the particular geometric models.

Take Away Points

- What is Configuration Space
- How Obstacle is Represented
- Two-Dimensional Robot and Obstacle
- Star Algorithm
- Obstacle Contour Formulation