CSCE 452/752 Fall 2025

17. Pursuit and Evasion

TEXAS A&M UNIVERSITY

AARRG

Motivation

Introduction

Pursuit-evasion tasks require a robot to find one or more **evaders** in its environment, or to verify that the environment contains no evaders.

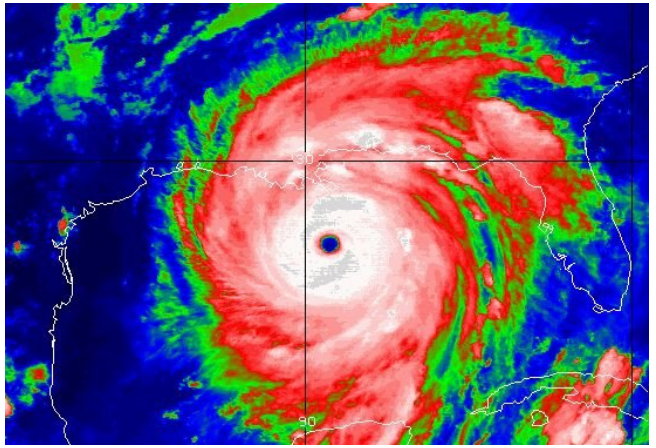
We can think of this as a sort of robotic “hide-and-seek” game.

In the bigger picture, this problem is a good example of how to design an algorithm to solve a nontrivial problem, for which the robot's **uncertainty** is a central part of the problem.

Who cares?

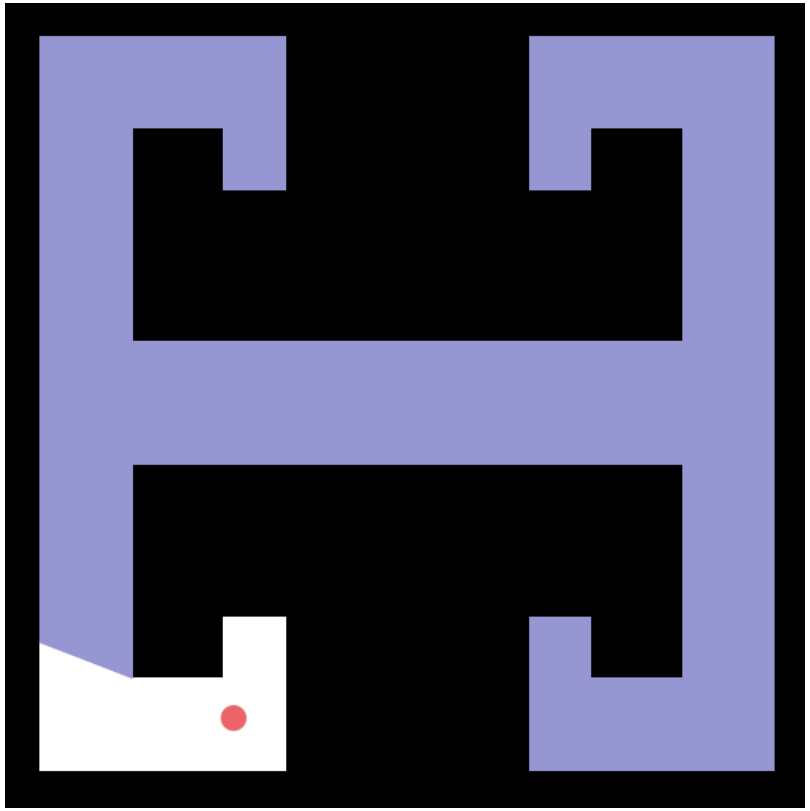
Good solutions to pursuit-evasion problems could be useful for:

- Patrolling a museum.
- Finding the “bad guys.”
- Finding victims in disaster situations.



The problem

A **pursuer** robot moves within a range sensor in a known, polygonal, planar, simply-connected environment. Its goal is to **see** an **evader** that moves arbitrarily quickly.

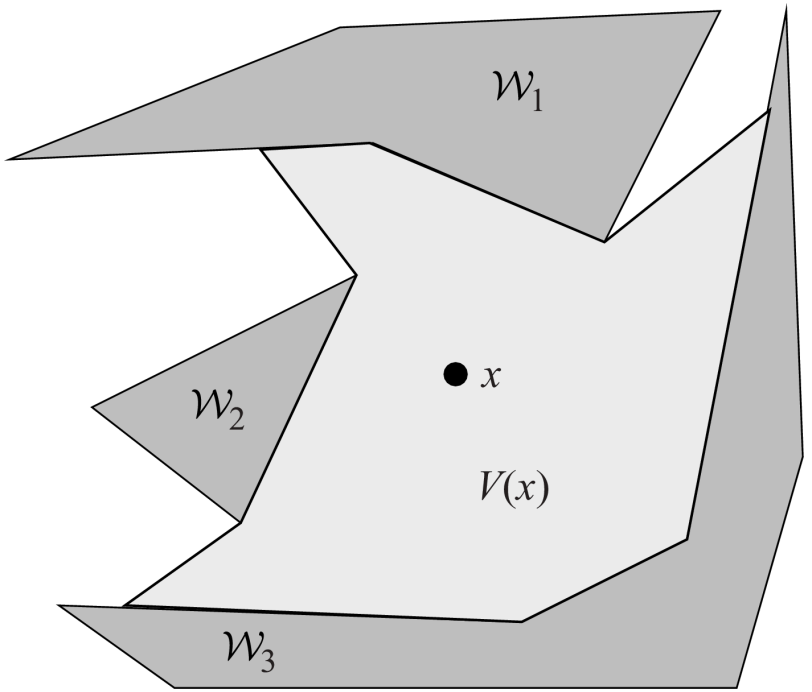


Gaps and labels

We can divide the boundary of pursuer's visibility polygon into **boundary edges** and **gap edges**.

Label each gap edge:

- If evaders might be hiding behind the gap edge, the gap is **contaminated**, labeled 1.
- If we are certain no evaders are hiding behind the gap edge, the gap is **clear**, labeled 0.

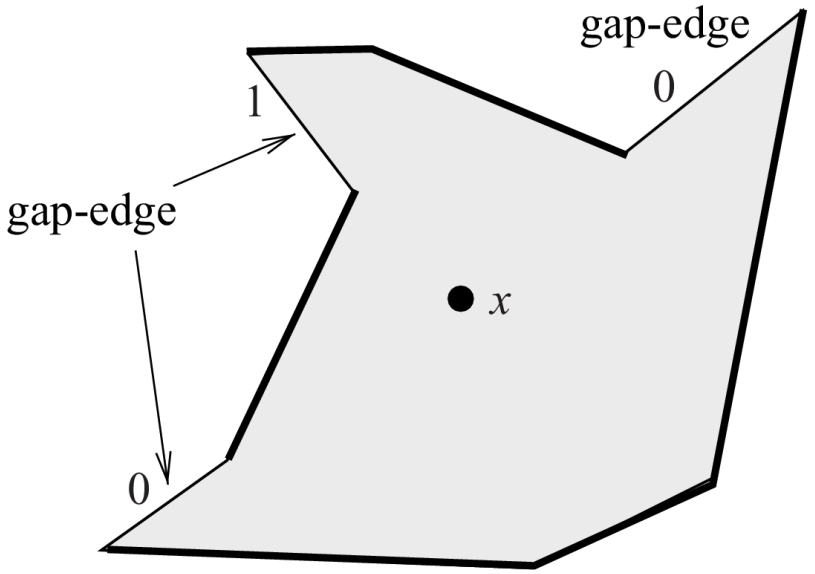


Gaps and labels

We can divide the boundary of pursuer's visibility polygon into **boundary edges** and **gap edges**.

Label each gap edge:

- If evaders might be hiding behind the gap edge, the gap is **contaminated**, labeled 1.
- If we are certain no evaders are hiding behind the gap edge, the gap is **clear**, labeled 0.

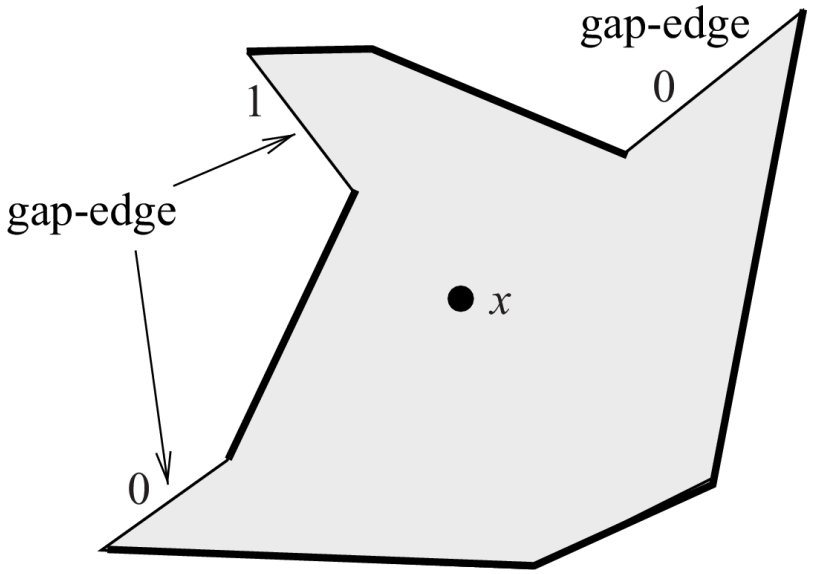


Gaps and labels

We can divide the boundary of pursuer's visibility polygon into **boundary edges** and **gap edges**.

Label each gap edge:

- If evaders might be hiding behind the gap edge, the gap is **contaminated**, labeled 1.
- If we are certain no evaders are hiding behind the gap edge, the gap is **clear**, labeled 0.



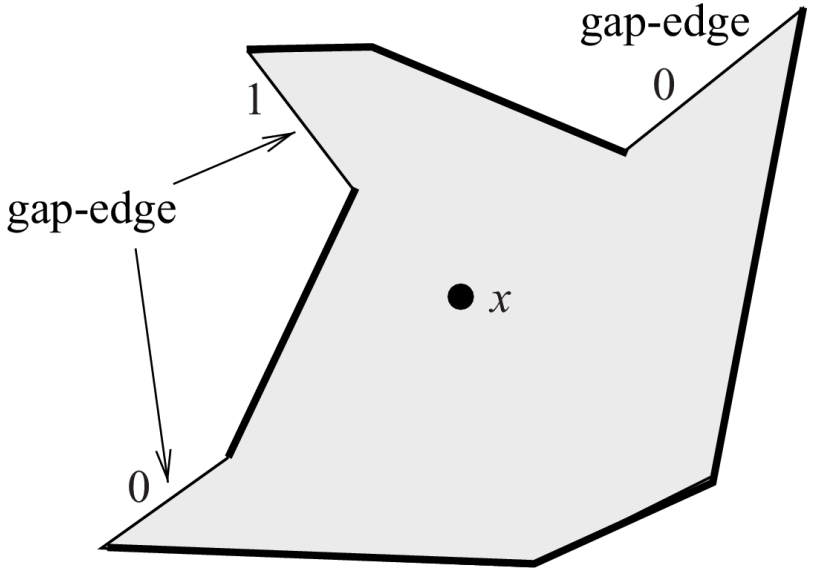
What are the initial labels?

Gaps and labels

We can divide the boundary of pursuer's visibility polygon into **boundary edges** and **gap edges**.

Label each gap edge:

- If evaders might be hiding behind the gap edge, the gap is **contaminated**, labeled 1.
- If we are certain no evaders are hiding behind the gap edge, the gap is **clear**, labeled 0.

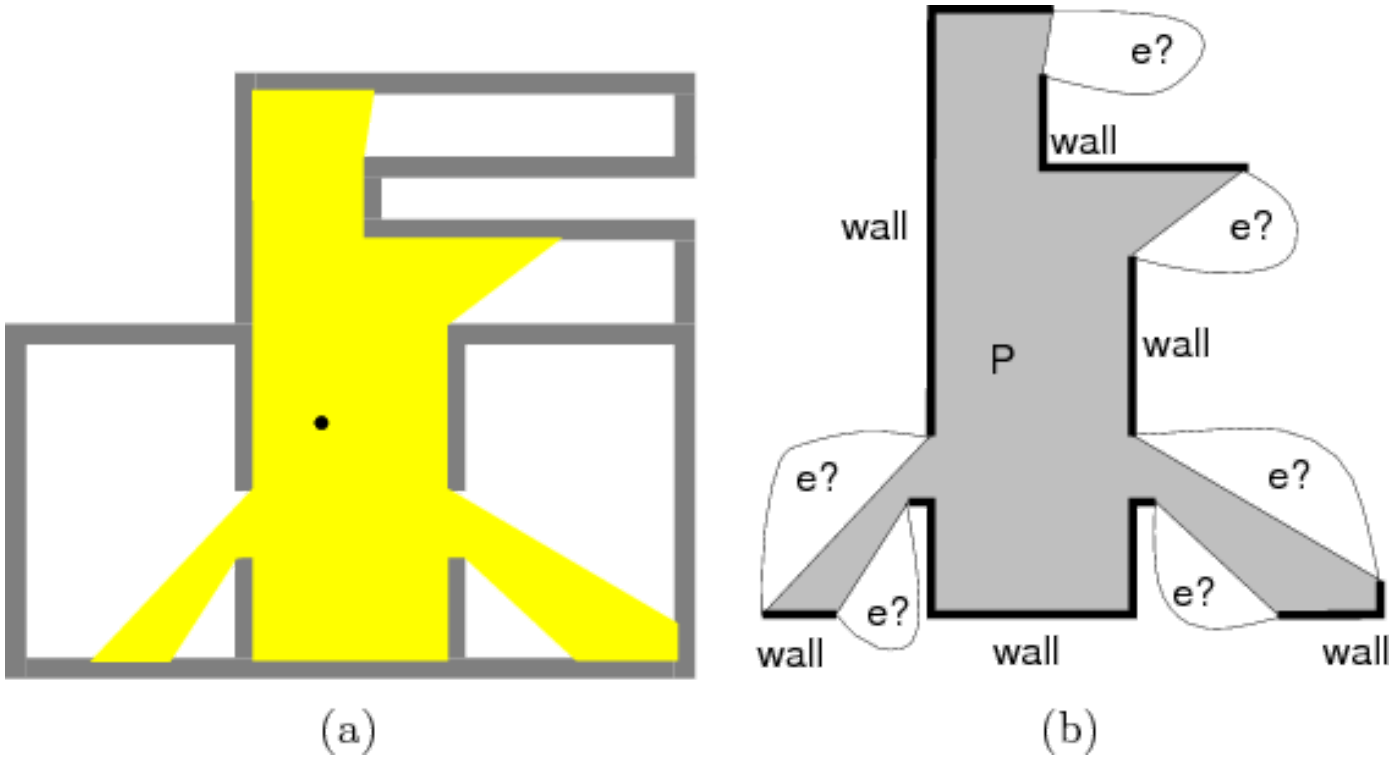


What are the initial labels?

What is the goal, in terms of these labels?

What does the pursuer know?

We can completely describe the pursuer's knowledge using these labels on the gap edges.



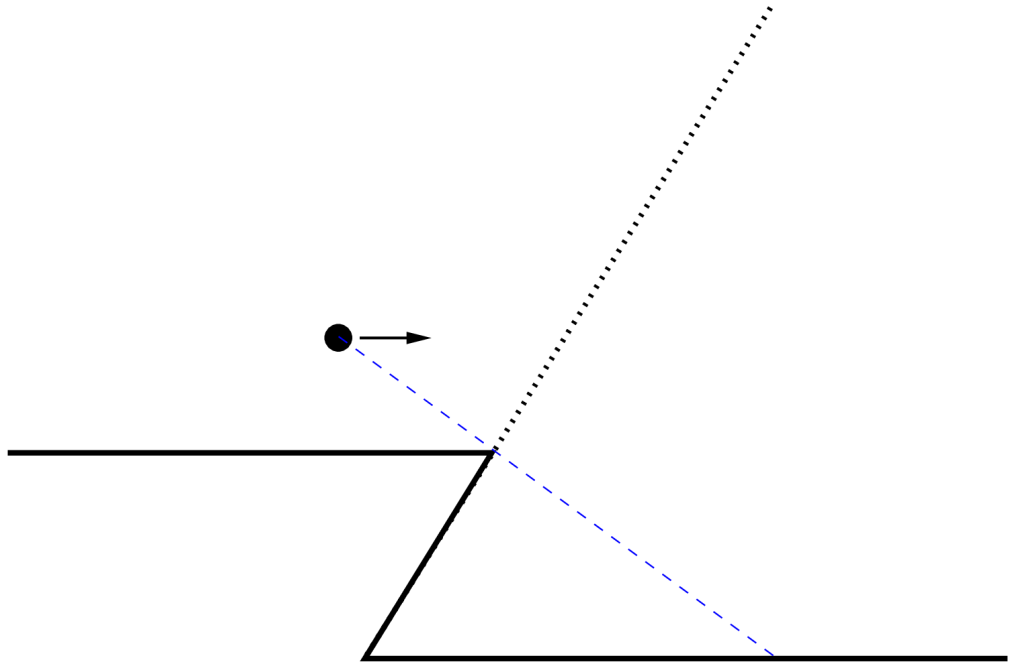
(a) (b)

Gap changes

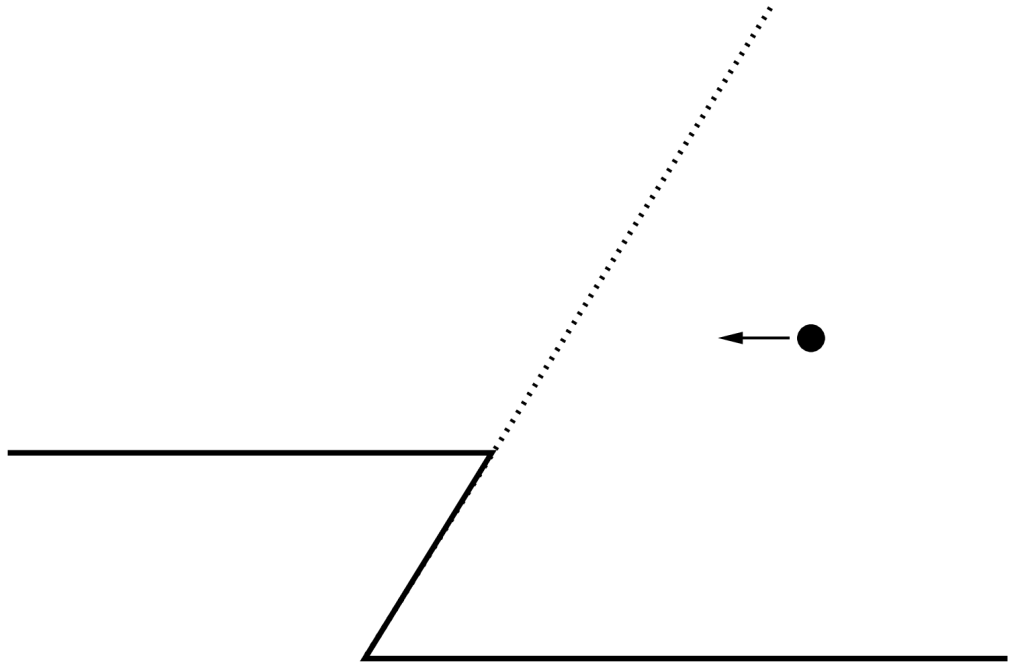
How can the gaps change as the purser moves?

- A gap can disappear.
- A new gap can appear.
- A gap can split into two gaps.
- Two gaps can merge.

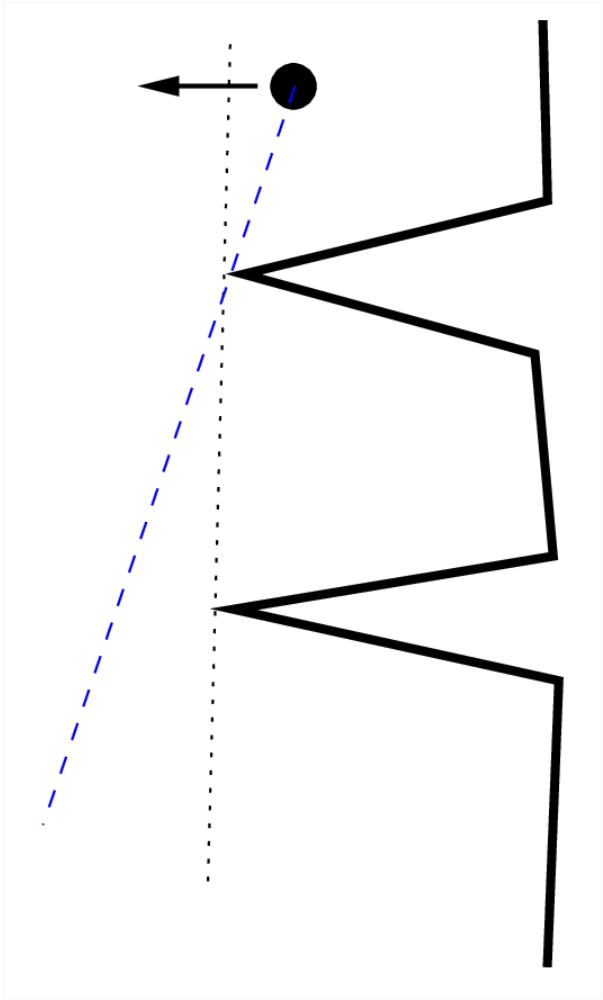
Disappear



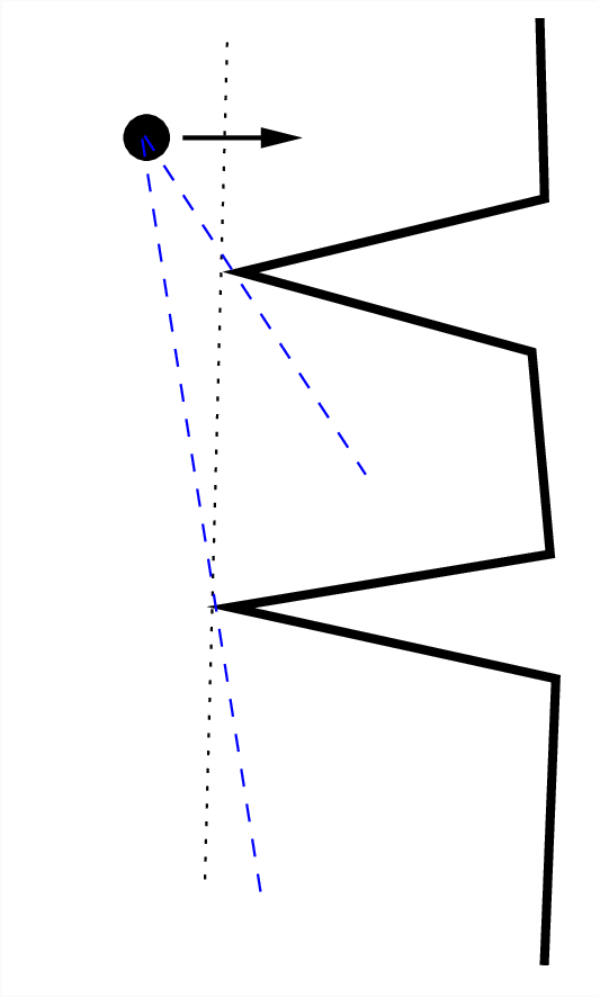
Appear



Split



Merge



Conservative regions

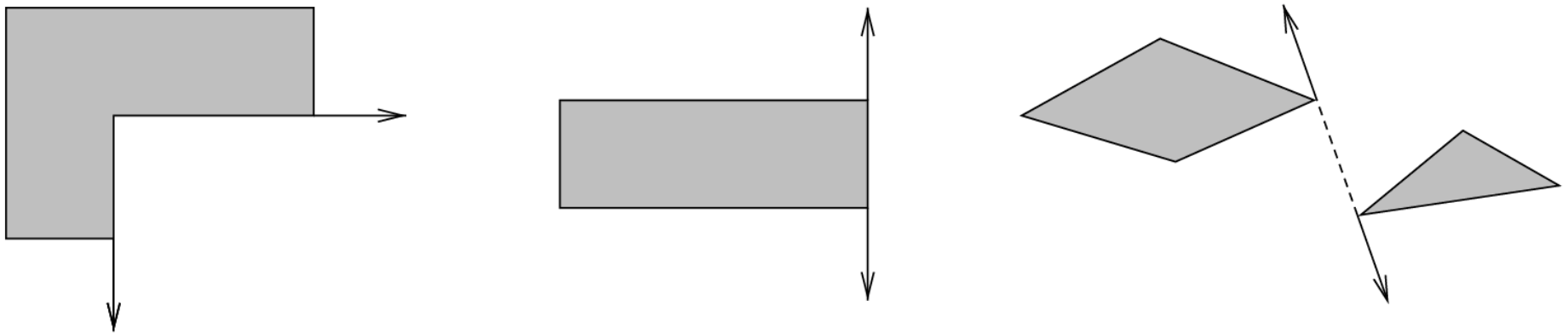
Unless it crosses one of the boundaries from the previous slides, the pursuer's gaps and labels remain the same.

The regions between these boundaries are called **conservative regions** because the robot can move freely through each cell without changing its knowledge.

Conservative regions

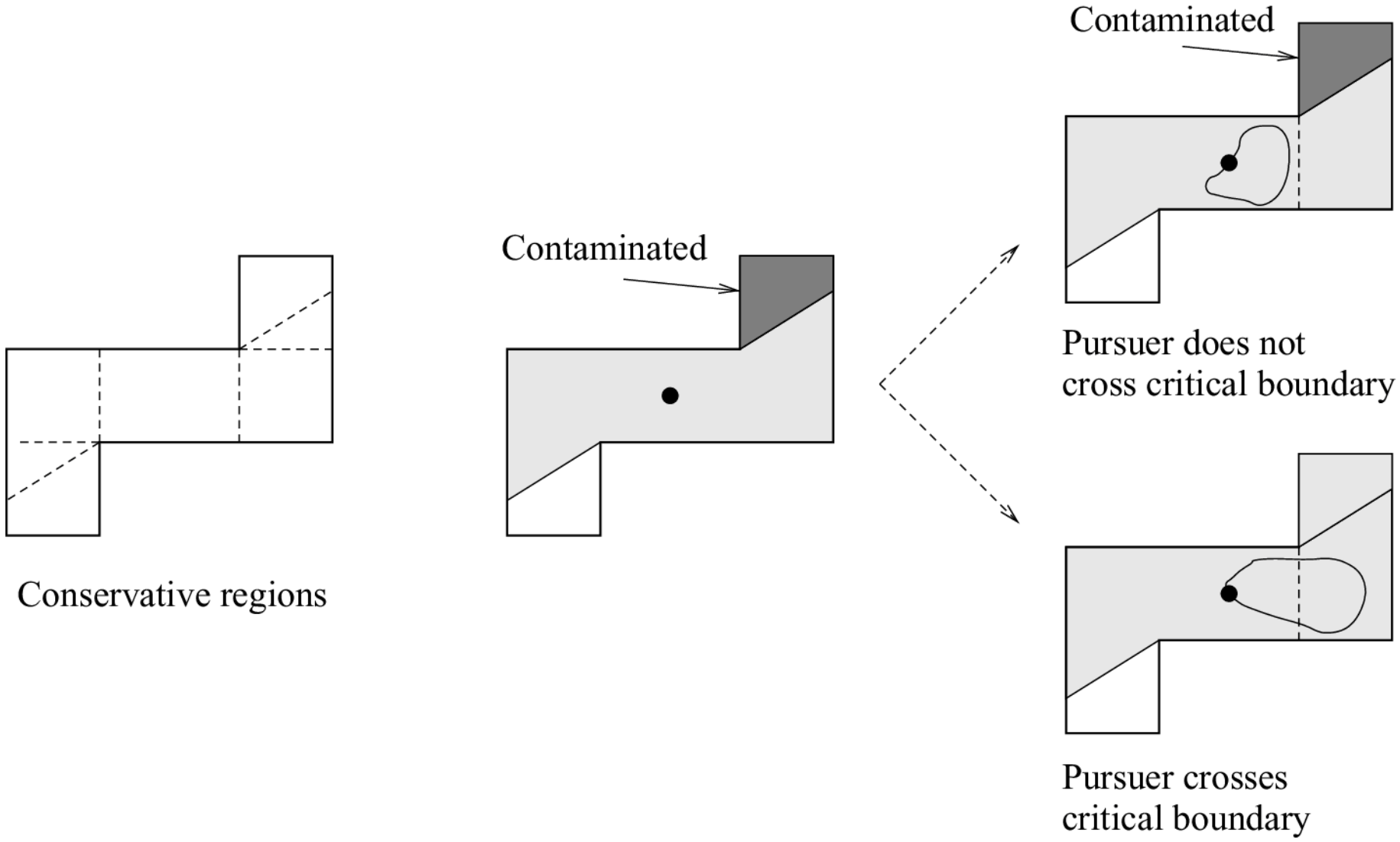
To divide the environment into conservative regions, draw rays:

- outward from the incident edges of each reflex vertex.
- outward from each pair of mutually visible reflex vertices.



Case 1 Case 2 Case 3

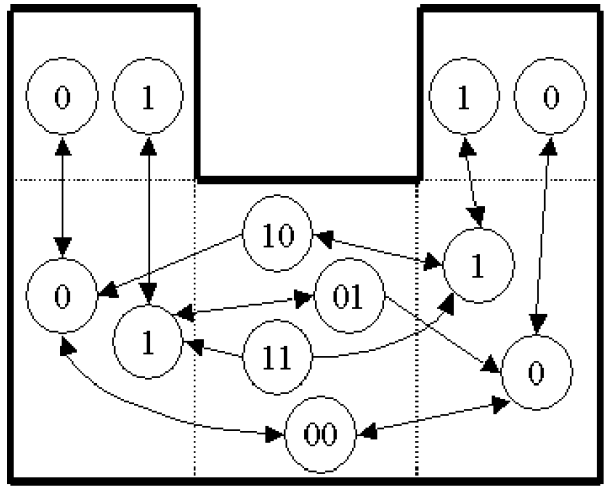
Conservative regions



The information graph

To solve the problem, we can form an **information graph**.

- Each node is a conservative region combined with a labeling of the gaps for this region.
  - If a conservative region has n gaps, there are 2^n nodes associated with it.
- Each edge is a directed transition between conservative regions, updating the labels appropriately.

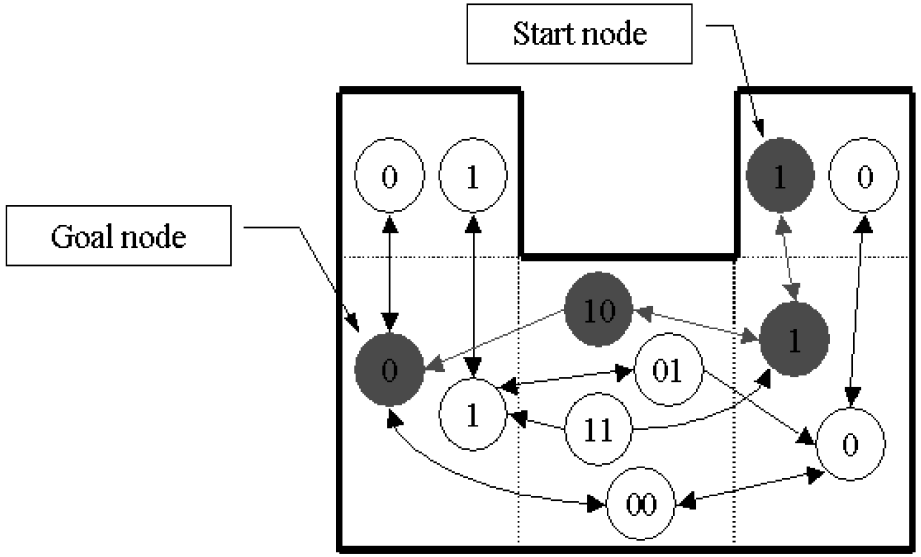


The information graph

To solve the problem, we can form an **information graph**.

This reduces the problem to a graph search on the information graph.

- From the starting node labeled with all 1's.
- Find a path to any node with labeled with all 0's.



Extreme example: Multiple recontaminations