Lab 2 and Homework 2

Moving Target

Problem Description

In this problem a point robot has to catch a moving target. Both the robot and the target live in an *N* x *N* 4-connected 2D grid-world, as depicted in Figure 1. This means they can only move up, down, left or right. Additionally, both the robot and the target can also decide to remain in the current position. However, neither the robot nor the target can move through the outer walls of the grid. Each cell in the grid-world is associated with the cost of visiting it. This cost of visiting one cell is 1, and the maximum cost is (*N*x*N*x*2)*. if the robot costs (*N*x*N*x*2)*, the game should stop and the planner should return a failure. Note that when the robot chooses to remain in the same cell, it will pay the cost of the visit again.

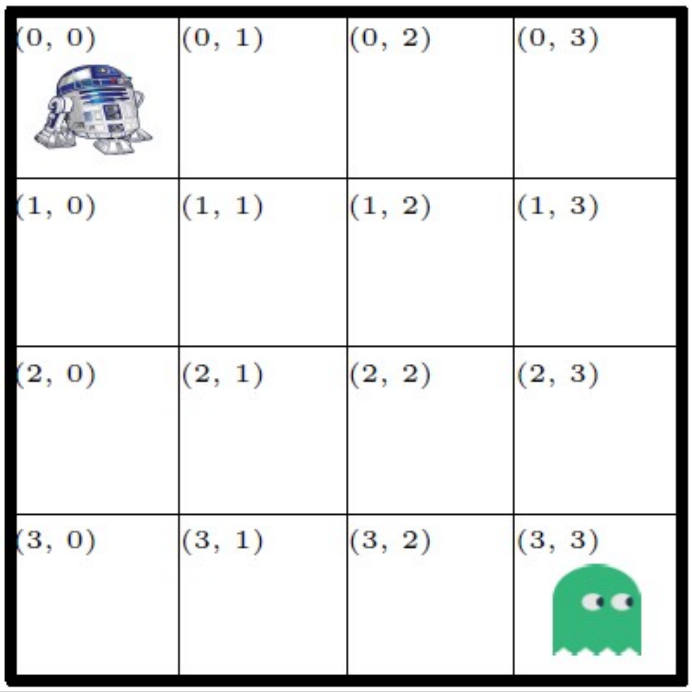

Figure 1: The robot and the target in the 2D grid-world.

The robot knows in advance the predicted one-way trajectory of the moving target, as a sequence of positions in the grid (for example: (initial) (3, 3), (2, 3), (2, 2) (final)). The first element of this trajectory is the initial position of the target at timestep *0* and the last element is the last position of the target at timestep *n*.

Both the robot and the target move at the speed of one cell per time-step.

[100 points]

In this assignment, you will be writing the required PDDL files for a planner to help the robot catching the moving target. The task of the planners is to generate the path of the robot should follow in order to catch the target. You could suggest a planner in x and y domain, but if you can implement it with x, y and t (time-step) domain, there a bonus degree for that.