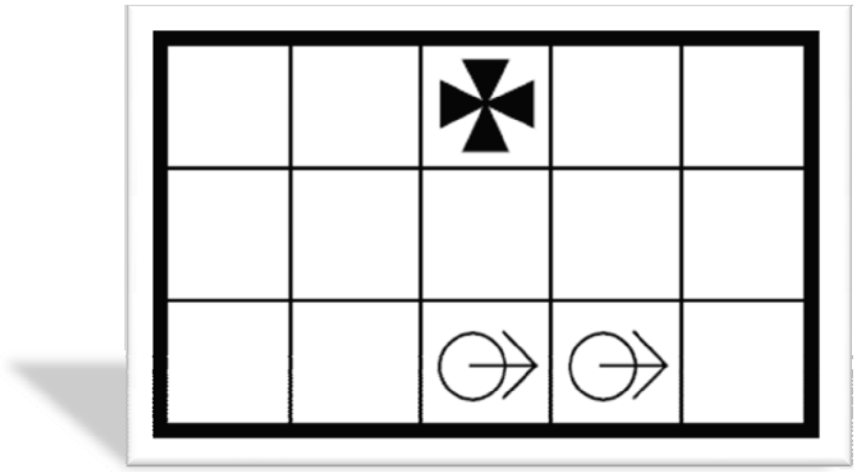
MISSION #9

Sequential Planning with Uncertainty

Due: September 2, 2010

Plain sequential programming was easy, but what happens in a case in which our robot has positional uncertainty?



The robot knows that it is either at ((2,0) r) or at ((3,0) r). The goal is to reach node (2,2). If the robot "assumes" one of the two positions and executes a corresponding sequential plan to the goal, it will end up at node (2,2) if its assumption was correct, or node (1,2) or (3,2) if its assumption was wrong. There is a sequential solution to the problem, though:

GTNN(1) GTNN(1) TurnTo(-90) GTNN(1) GTNN(1) TurnTo(-90) GTNN(1) GTNN(1)

This succeeds because of the behavior of GTNN when the robot is facing a wall. Therefore, after the first two moves, the robot is definitely at ((4,0), r). This is an example of coercion, a movement that collapses the number of possible states.

Assignment 5.1: Sequential Planning with Uncertainty

Task: Write a robot function called ***SPU()*** *that returning a plan*

Description: Write an uncertain sequential planner. When called, this planner attempts to find the shortest sequential plan no longer than ***maxDepth*** that would result in the robot reaching a goal state if its initial state were any state consistent with the initial conditions. If there is a solution, this function should return the solution plan. Otherwise, the function should so indicate. You may assume that the map is complete but the robot position specification will be a set of possible positions. Your robot should then take the resulting plan and execute it!

As with ***Assignment 5.0***, you will need a progression function to determine the result of applying an action to the states in a state-set. The function from ***Assignment 4.2*** does exactly this.