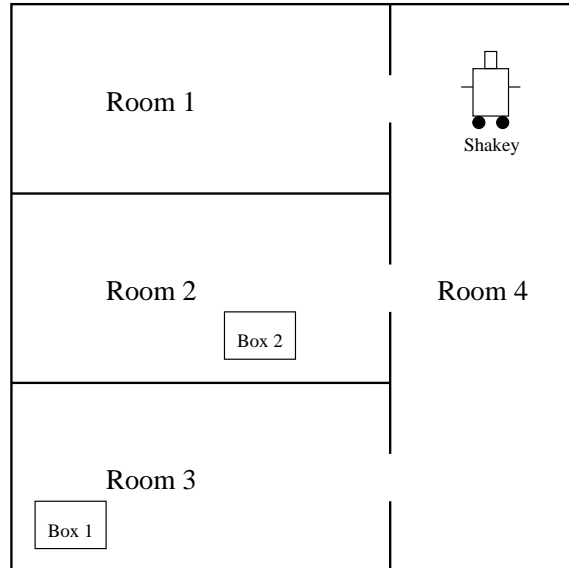


Question 1. (based on Russell & Norvig, pp. 414–415, 11.13¹) [20 points]

Now it's your turn to program Shakey. Below is the layout of Shakey's world (at least for this problem). There are four rooms, three doors and two boxes (that Shakey can push). There are a number of different locations. Assume that a door represents a location that is in both of the adjacent rooms. There are also other (physical) locations in the rooms (unspecified on the map below).



Shakey can do two things (in this limited world): he can GO to a location (from another location), provided the two locations are in the same room, and he can PUSH a box from one location to another location (provided he is already at the box's location and the two spots are in the same room).

This problem lets you be the designer of who to represent this in a planning domain. As such, you will need to make certain other assumptions and design choices. Be reasonable about it. Assume we'd like to use the two functions above to drive Shakey. Someone has already implemented a low-level controller that will take Shakey from one Cartesian coordinate to another, provided both are in the same room. Assume that there are already sets of coordinates pre-defined as locations in each room to which Shakey can drive (both boxes' and Shakey's current position being examples).

part a. First let's approach this with situation calculus. Write down the axioms for Shakey's world, including the initial condition, "possibility" statements, definitions for the change in fluents, and any other axioms necessary to plan.

part b. Shakey's goal is to get box 2 into room 1. Write this down in situation calculus.

part c. Now let's approach this with STRIPS. Write down the initial state and Shakey's two actions in STRIPS-notation.

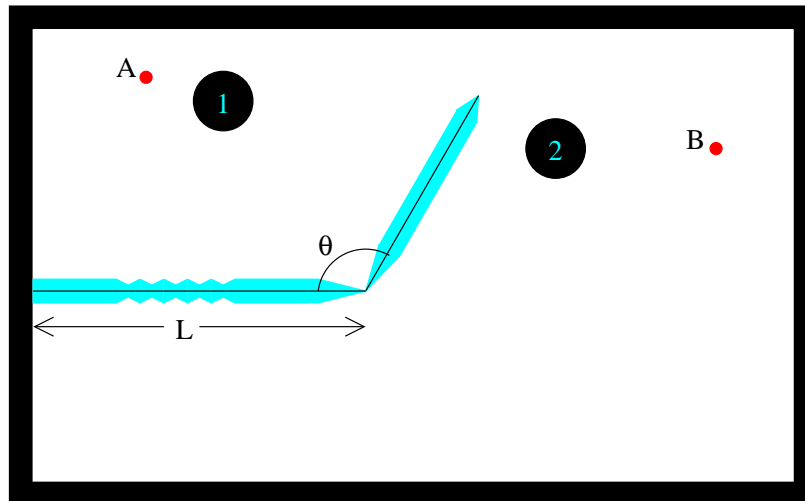
part d. Write down Shakey's goal, but now in STRIPS.

part e. Give an example of a goal in this world which can be expressed in situation calculus, but not in STRIPS.

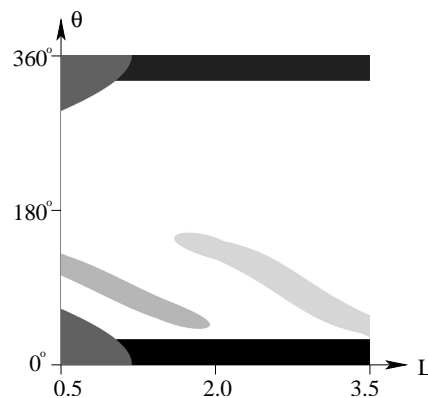
¹11.13 in the text has a few errors. Additionally, I have modified the problem slightly. Please do not work directly from the book problem or you may be led astray.

Question 2. [20 points]

Consider the robotic arm shown below.² Its “lives” in a 2-dimensional box with two immobile circles (marked 1 and 2) blocking its workspace. It has two joints. The first is a prismatic (telescoping) joint whose length we denote by L . The stops on the first joint limit L to the range between 0.5 and 3.5 inches. The second is a revolute joint whose angle we denote by θ , as marked. Ignore the points marked A and B for the moment.



Below is the configuration-space for this robotic arm. The white area is free space; the shaded areas represent obstacles.



- Label each of the shaded regions with the obstacle or constraint it represents in the workspace.
- Two constraints or obstacles have been omitted from the above configuration-space. Add and label them.
- We want the arm to move the end of its last link from point A to point B in the workspace. For both of these points:
 - Find an arm configuration which reaches the point and 2. Plot this arm configuration on the configuration-space diagram.
- Draw a free path in the configuration-space between the two configurations from part c. It doesn't have to be optimal with respect to anything in particular, but it should be reasonable.
- Select four representative points (other than the end points), roughly equally spaced, from the path in configuration-space and label them in configuration space. Then draw the arm's position (two lines representing the center lines of the segments is fine) for each of these four points in the workspace.

²The two figures in this problem are available as postscript files on the course website.

Question 3. [60 points]

Your friend Ralph once told you “Build a better robber trap and the world will beat a path to your door.”³ With all of your AI experience, you should have no problem. And, naturally, your biggest market segment is museums. So, you will build an Automated Robber Tracking System because the acronym will appeal to museum directors.

All museums are laid out on a grid.⁴ Some grid cells are occupied by solid walls (and therefore it is very unlikely to find a robber there). The others are open for people to walk around. Because robbers are people too, they use the same space for their evil doings. So, all museums have robber sensors⁵ in each grid cell.

That’s the good news. The bad news is that these sensors do not always work properly. Sometimes they report that a robber is there when he really isn’t and other times they fail to report a robber when there really is one at the sensor’s location. Sensor failures are completely independent of each other and transient (independent of other failures of the same sensor at different times). The worse news is that the sensors take a lot of electrical power to run. It is not generally possible to run all of the sensors in the museum at the same time. You have a power budget that dictates the number of sensors you can have on at one time.

The only mitigating factor is that your crack psychology team working on this project have managed to formulate a highly sophisticated model of how robbers move around inside of museums. They have determined, after years of study, that robbers always enter a museum from the outside (a grid cell on the boundary of the museum), facing a random direction. At any given moment in time, they either turn (change direction randomly by 90 degrees) or move forward. If they are facing a wall, they always chose to turn. Otherwise, they turn with a fixed probability. The money you spent sponsoring this research was clearly well spent.

Your goal is to use this detailed knowledge of museum robber psychology and your incredible hidden Markov model expertise to design a system capable of cutting into that all-important economic market of museum security. While there are a number of ways of tackling this problem, here is the method you should employ. Your system will view the world as a hidden Markov model (HMM) in which the state of the system will be the current position and orientation of the robber.

The robber enters the museum at the first time step. Before each movement of the robber, your guard will be given the chance to select a number of sensors to turn on. After turning them on, the sensor values will be reported back to the guard. Based on this and the fact that the robber moves (or rotates) once between sensor readings, you can track a distribution over the robber’s state. When asked which sensors to turn on, your guard should select the grid cells which have the highest probability of the robber being there (according to your current estimate). At the end of a run, your guard will be asked to predict the path that the robber took. There are many ways of interpreting this query mathematically. You should select, for each time step, the single grid cell with the highest probability that the robber was there, given *all* of the evidence (before and after).

The starter code kit on the web contains code that keeps track of probability distributions and conditional distributions. The former isn’t too difficult. However, the latter (at least in this project) needs to be stored in a sparse format to make the calculations quick (hopefully in working through the assignment, it will become clear why this is true). Therefore, they are being supplied. You should use these in your project. The maze class describes the layout of the museum and will convert any free location in x, y coordinates into a unique index, and vice-versa (which should be helpful in describing distributions).

Four museums are provided as input text files. Feel free to create your own (simpler) ones for testing. When you have completed the coding, for each of the provided museum plan files, provide the empirical average and standard deviation of the number of correct predictions and the average distance error, averaged across 20 runs of your algorithm.

³Of course, he didn’t quite use the word “robber;” but then again he also said, “Foolish consistency is the hobgoblin of small minds,” and what’s more foolish than letting a little consistency get in the way of a good story? I know Ralph would agree.

⁴In fact, despite your experiences in the real world that may lead you to erroneously conclude otherwise, all such grids are rectangular.

⁵A typical robber sensor consists of a patented design of motion detectors, laser beams, and other things that film well for movie special effects.