Zewail City of Science and Technology
Space and Communications Engineering
SPC418: Autonomous Vehicles Design and Control
Fall 2016-2017

Assignment-3

Due date: December 15, 2016 at 11:59PM

Notes:

- This assignment is a group assignment. The group SHOULD be same group as course project.
- Make sure you installed ROS and configured your workspace as explained in the second tutorial.
- Get CatBot ROS package from SPC418 GitHub Repo: https://github.com/spc418-ZC/SPC418-Fall-2016
- Anything handed in after the due date will be penalized by 50% for each 24 hours of lateness.

What to submit: a report that contains:

- A short assignment report reporting your observations and screenshots of the implemented code, typed or neatly handwritten.
- ROS-compatible code for all the included programming exercises.
- Zip the assignment report and the source code (including a README file) and name it "**Assignment3-Your Project#.zip**".
- Send this file to the course TA, Ahmed Khairy.

1. **[Written Exercise – 1 Mark]** Robot rental is a fast growing business. Nowadays many companies rent robots for different applications. For example, the humanoid robots "Wakamaru" designed by Mitsubishi Heavy Industries Ltd are being hired out in Japan to serve as receptionists at offices and hospitals for about 120,000 yen ($1000) per day, or at about 3 million yen ($25,000) a year. This robot can recognize about 10,000 words necessary for daily life and is able to identify/recognize visitors and even conduct a simple conversation with the customers.

Assume that a robot company rents 40% of the robots for its customers from agency-I and 60% from agency-II. If 6% of the robot from agency-I and 5% of the robots from agency-II break down. What is the probability that a robot rented by this company breaks down?

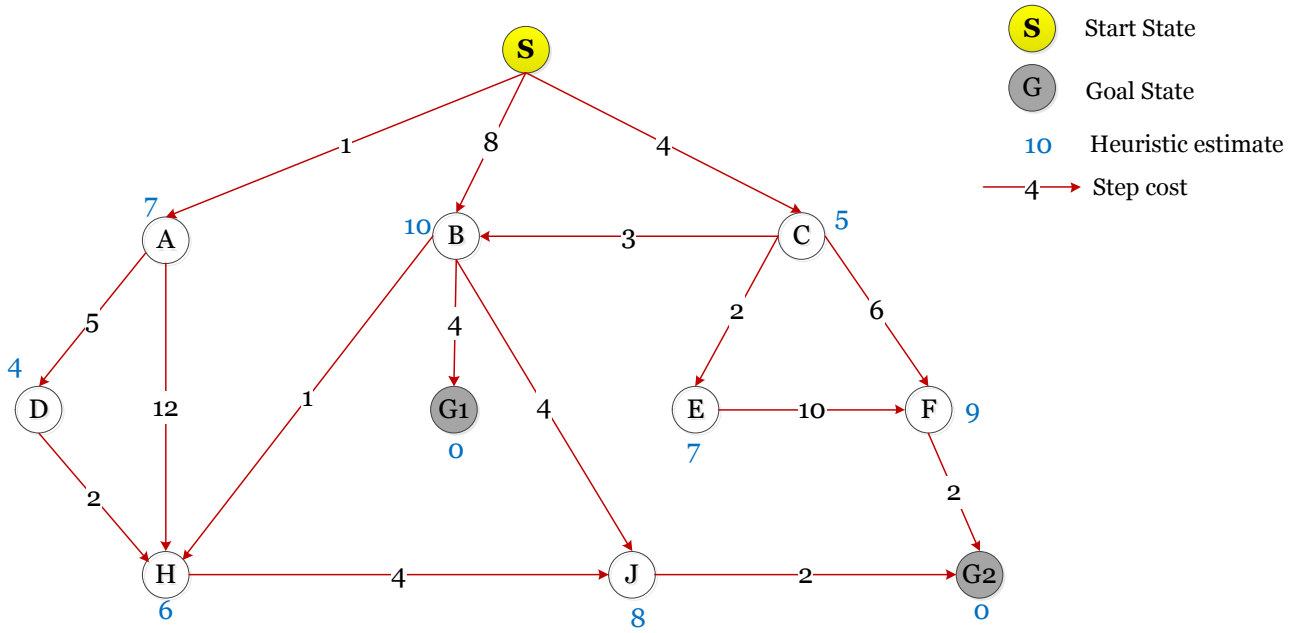
Mitsubishi Wakamaru

2. **[Written Exercise – 2 Marks]** Consider the search space below, where S is the start node and G_1 and G_2 are goal nodes. Arcs are labeled with the value of a cost function; the number gives the cost of traversing the arc. Above each node is the value of a heuristic function; the number gives the estimate

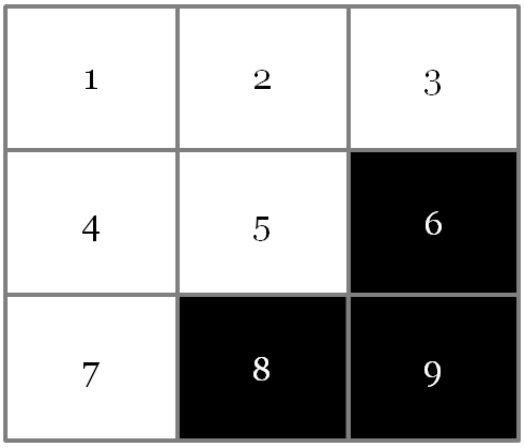
of the distance to the goal. Assume that forward search algorithms always choose the left branch first when there is a choice. For each of depth-first search (DFS), breadth-first search (BFS), and A* search strategies,

(a) indicate which goal state is reached first (if any) and

(b) list *in order*, all the states that are popped off the OPEN list.

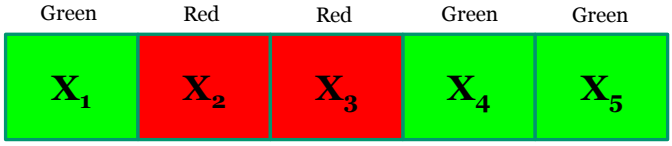


3. **[Written Exercise – 2 Marks]** A mobile robot equipped with four 1-D Gaussian sonar sensors (Front, Right, Left and Back) with zero-mean and standard deviation=$0.02\rho^2$ where ρ is the actual range to the target. If this robot is navigating in the environment as shown below starting from cell 1.



Show how the robot can build a map for this environment using 3 ultrasound scans in three different positions (cell 1 as initial position, cell 3 and cell 5).

4. **[Programming Exercise – 2 Marks]** If the CatBot can be in one of five grid cells, labeled X_i, for $i = 1 \ldots 5$. Two of these cells (X_2 and X_3) are colored red, and the other three (X_1, X_4 and X_5) are green as illustrated below.



a) Calculate the robot's initial belief about its location in the world (the probability for each X_i)?

b) The CatBot's initial belief calculated in a) can be updated given a measurement from the robot's sensors. The following simple rule is to be used to represent the probability that the robot is in a red or a green cell, based on the robot's measurement of "red":

pHit = 0.6 → factor for matching measurement

pMiss = 0.2 → factor for non-matching measurement

Update the code to calculate the probability that the robot is in a red cell and the probability that it's in a green cell?

c) Calculate the sum of the updated probabilities of the five cells and report your observation.

d) Let's make our code more elegant by introducing a variable, **world**, which specifies the color of each of the cells -- red or green, i.e., world = ['green', 'red', 'red', 'green', 'green']. Define a function to be the measurement update called **sense**, which takes as input the **initial distribution** p and the **measurement** Z (i.e, Z = 'red'). Enable this function to output the non-normalized distribution of q, in which q reflects the non-normalized product of input probability (0.2) with pHit or pMiss in accordance with whether the color in the corresponding world cell is red (hit) or green (miss). This function should work for any possible Z (red or green) and any valid initial probability distribution.

e) Modify this code so that it normalizes the output for the function sense, and adds up to one.

f) Try typing *green* into your **Z**, sensory measurement variable, and re-run your code to see if you get the correct result.

g) Modify the code in such a way that there are **multiple measurements** by replacing Z with a measurements vector. Assume the robot is going to sense red, then green. Can you modify the code so that it updates the probability twice, and gives you the **posterior distribution** after both measurements are incorporated so that any sequence of measurements, regardless of length, can be processed?

5. **[Programming Exercise – 3 Marks]** As a continuation for the programming exercise in Assignmnet-2:

a) Modify the implemented *kinematic* node of CatBot by adding three additional arguments to the driving function. These arguments are **x_noise**, **y_noise** and **turn_noise** that represent the amount of noise in the drive motion and in the turn motion respectively. These noises are uniformly distributed. For example, if you set the **turn_noise** to 0%, and execute a command turn(PI/2), then the robot will turn 90 degrees counter clockwise. If on the other hand, your **turn_noise** is set to 20%, then a turn(-2*PI*100/360), which is equivalent to a 100 degree turn clockwise, will result in a turn between -90 to -110 degrees. The actual value is chosen with uniform distribution from this range. Similarly, a drive with x=-1000mm and y=0 command with a **x_noise** and **y_noise** of 10% and a **turn_noise** of 20% will result in the robot driving with uniform distribution between -900 to -1100mm.

b) Without considering the noise parameters **x_noise**, **y_noise** and **turn_noise** and assuming that the robot is driving in a straight line along the x axis, starting at the true position x=0. The robot executes driving commands with distance *d*, where *d* is an integer, and it receives sensor data from its on-board global (absolute) positioning system *z*, where *z* is also an integer.

c) The robot's driving accuracy from an arbitrary starting position has to be established by extensive experimental measurements and can then be expressed as Gaussian distribution with zero-mean and standard deviation=$0.02d^2$ where d is the actual driving command. Similarly the accuracy of the robot's position sensor was established by measurements, before it can be expressed as Gaussian distribution with zero-mean and standard deviation=$0.01r^2$ where r is the true position.

Assuming the robot has executed the following sequential commands. After completion of each command, robot local sensor reports the corresponding position shown below:

Driving command (d)	2	1	1	2	1	1
True position (r)	2	2	3	4	5	5

Implement the node **bayes** that receives the driving commands and the sensor measurements and returns the belief vector that contains the estimated state of the robot position after each command.