CSL302– Artificial Intelligence
Lab 4

Due on 19/4/2018 11.55pm

Instructions: Upload to your moodle account one zip file containing the following. Please do not submit hardcopy of your solutions. In case moodle is not accessible email the zip file to the instructor at ckn@iitrpr.ac.in. Late submission is not allowed without prior approval of the instructor. You are expected to follow the honor code of the course while doing this homework.

1. **You are allowed to work in teams of size at most 2 for this programming lab**.
2. **Please use Python or C++ as the language of implementation**.
3. A neatly formatted PDF document with your answers for each of the questions in the homework. You can use latex, MS word or any other software to create the PDF.
4. Include a separate folder named as 'code' containing the scripts for the homework along with the necessary data files.
5. Include a README file explaining how to execute the scripts.
6. Name the ZIP file using the following convention rollnumber1rollnumber2hwnumber.zip

1. Shakey the Robot

In this lab you will implement below planning algorithms for the Shakey's world

- Forward (progression) planner using breadth first search
- Goal Stack planner

The Shakey's world is described as follows:

Shakey's world consist of N rooms where each room has a light switch and B boxes. It can move from one room to another, push boxes from a room to another, climb on (and down from) boxes and turn light switches on and off. It needs to climb on a box to turn on or switch off a light switch, and we assume that all rooms are connected by all other rooms ie it can directly go from room 1 to room 3. At a time, Shakey can push only one box from one place to another.

The propositions for this problem are as follows:

(at object location) - tells the room in which object(Shakey or boxes) is present.
(on shakey b) - Shakey is on b. Here b could be floor or box
(switchon light) - tells light is turned on

Note *shakey and floor are used as constant symbols*.

Note that these are propositions, so you will have to exhaustively list out the propositions that describe the current state. There are 6 actions specified using the following schemas:

go (x y)
preconditions – (at shakey x) (on shakey floor)
effects – ~(at shakey x) (at shakey y)

push(b x y)
preconditions – (at b x) (at shakey x) (on shakey floor)
effects – ~(at shakey x) (at shakey y) ~(at b x) (at b y)

climbup(b x)
preconditions – (at b x) (at shakey x) (on shakey floor)
effects – (on shakey b) ~(on shakey floor)

climbdown(b)
preconditions – (on shakey b)
effects – ~(on shakey b) (on shakey floor)

turnon(l x)
preconditions – (at l x) (at b x) (at shakey x) (on shakey b) ~(switchon l)
effects – (switchon l)

turnoff(l x)
preconditions – (at l x) (at b x) (at shakey x) (on shakey b) (switchon l)
effects – ~(switchon l)

Please note in above schema, (x,y) could be only rooms, (b) could be boxes and (l) will be the light switch in every room. (I could have modelled these constraints using three more propositions: location(x) - x is a room, box(b)-b is a box and light(l)- l is a light . But it would have resulted in a more complex definition of the state.)

For the sake of simplicity, you can explicitly encode these actions in your code. A state will be specified as a list of propositions that hold good in that state separated by a space in a single line. For example, in Shakey's world, a state can be (at b1 r1) (at shakey r1) (on shakey floor)

Given a text file containing the initial and goal state description, and the choice of the planning approach, your software should output a file containing the plan from the initial state the goal state. Some sample problems are given in the folder p1 of the lab.

Input

The input to your code will be the name of the text file a description of the initial and goal states along with the planning approach that has to be used. Specifically, the format of the input file is as follows

N B
planner
initial
State description
goal
State description

In first line, N is the number of rooms and B is number of blocks. The second line indicates the choice of the planner (f-forward planner with BFS and g- goal stack planner). The third line indicates that the line following it contains the complete description of the initial state. This is followed by the line containing the term goal. This is in turn followed by the line that completely describes the goal state.

Output
Your code must output to a text file the plan specified in the following format
NA
Action 1
Action 2
...
Action NA

The first line indicates the number of actions in the plan. Each line then presents the action that has to be taken.
Compare and contrast the different search techniques in terms of the number of nodes expanded and the length of the plan. Included as part of the assignment are sample problems describing the initial and goal states.

Sample Problem:

2 1
f
initial
(on shakey floor) (at b1 r1) (at shakey r1) (switchon l1)
goal
(on shakey floor) (at b1 r1) (switchon l2)

Solution:

(push b1 r1 r2)
(climbup b1 r2)
(turnon l2 r2)
(climbdown b1)
(go r2 r1)

2. Probabilistic Inference using Bayesian Networks

In this lab you will implement Gibbs sampling for drawing inference on the probability of a sample from a Bayes Net. We will use the Adult income dataset [1] for this part of the lab. We have included the train and test samples of this dataset as part of the accompanying folder p2. Our objective is as follows – use the training set to construct a Bayes net that models the full joint distribution. For every instance in the test set, use the constructed Bayes net to draw inference on the likelihood of the instance being drawn from this distribution. To make your lives simple, we have also included an implementation of the Chow-Liu algorithm as part of the lab in folder p2. This python script will create the Bayes net and compute the conditional probability distribution table at every node of the network. The Chow-Liu algorithm constructs a minimal spanning tree of the complete network based on mutual information in the features. Refer to [2] for more details about the Chow-Liu algorithm.

Your objective is to perform Gibbs sampling on this network to draw inference on the likelihood of a test sample being generated by the full joint distribution as encoded by the Bayes net. Implement a Gibbs sampler that can draw samples from the Bayes net. Vary the number of Gibbs sampling iterations to draw the sample, and the total number of samples drawn by the Gibbs sampler to study the goodness of the likelihood estimate. Compute the population estimate of a sample from the test set and use it as the ground truth to compare against the estimate obtained through Gibbs sampling. What are your observations on the relationship between the accuracy of the estimate, the number of samples drawn from the Bayes net by the Gibbs sampler, and the number of Gibbs sampling iterations?

[1] https://archive.ics.uci.edu/ml/datasets/adult
[2] Chow, C. K.; Liu, C.N. (1968), "Approximating discrete probability distributions with dependence trees", IEEE Transactions on Information Theory, IT-14 (3): 462–467.