CSL302– Artificial Intelligence
Lab 4

Due on 7/4/2017 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. **This lab has to be worked upon individually**.

2. 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.

3. Include a separate folder named as 'code' containing the scripts for the homework along with the necessary data files.

4. Include a README file explaining how to execute the scripts.

5. Name the ZIP file using the following convention rollnumber1rollnumber2hwnumber.zip

---

1. Planners for Block World

In this lab you will implement the two planners designed for the blocks world problem that we discussed in the class.

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

The blocks world is described as follows:
There are N blocks, table and a robotic arm. Blocks are identified by integers 1 to N. Each block can sit on top of another block or on the table. There can be a stack of blocks of arbitrary height. However only one block can be directly on another block. No two blocks can be sitting directly on the same block. The bottom most block of a stack must be on the table. The table can hold any number of blocks. If there is no block on top of a block, then the block is clear. The robotic arm can hold only one block. If the robotic arm does not hold any block, it is empty.

The propositions for this problem are as follows:
(on blocka blocka) – meaning blocka is stacked on blockb
(ontable block)
(clear block)
(hold block)
(empty)

Note that these are propositions, so you will have to exhaustively list out the propositions that describe the current state. There are 4 actions specified using the following schemas:

action(pick block)
preconditions – (ontable block) (clear block) (empty)
effects – (hold block) ~(clear block) ~(empty) ~(ontable block)

action(unstack blocka blockb)
preconditions – (on blocka blockb) (clear blocka) (empty)
effects – (hold blocka) clear(blockb) ~(on blocka blockb) ~(empty) ~(clear blocka)

action(release block)
preconditions – (hold block)
effects – (ontable block) (clear block) (empty) ~(hold block)

action(stack blocka blockb)
preconditions – clear(blockb) (hold blocka)
effects – (on blocka blockb) (clear blocka) (empty) ~(hold blocka) ~(clear blockb)

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 a 3-blocks world, a state can be (on 1 2) (clear 1) (ontable 3) (ontable 2) (clear 3) (empty).

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. You have define a good heuristic for performing A* search based forward planner.

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
planner
initial

State description
goal
State description

The first line is the number of blocks in the blocks world. The second line indicates the choice of the planner (f-forward planner with BFS, a- forward planner A* search 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.

Also include a pdf file that describes the heuristic function that you defined for the A* search. 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.

Please use Python or C++ as the language of implementation.