CS 309
Autonomous Intelligent Robotics (FRI I)
HW 3: ROS Basics, Some Simple Computer Vision and Some Quick PDDL
Due: March 24, 2020

You can find the latest version of this PDF at
http://justinhart.net/files/homework/2020_spring_cs309/HW3/hw3.pdf

You can find the rosbag used in this homework at
http://justinhart.net/files/bags/three_cups.bag

The first part of this homework is an exercise in computer vision, in which you will use color segmentation to pick brightly-colored cups out of a video stream. Your implementation should compile using catkin_make or catkin build with g++ under ROS Kinetic and Ubuntu 16.04.

The second part of this homework is to implement the Towers of Hanoi in PDDL. Your PDDL should run under FF.

ROS and OpenCV

- In class, we learned how to extract an image of a blue cup from a rosbag using OpenCV. That rosbag can be found at the URL above. In the video in that bag, there are 3 cups. Your job is to stream a ROS topic that shows only those three cups, but shows all three cups. To do so, you will modify the code example from that lecture, which can be found in the examples on justinhart.net. Your code should:
  - Extract the blue cup using color subtraction and thresholding methods as outlined in class. It should use color channel subtraction, thresholding, contour detection, and masking to perform this task. Comments are not mandatory, but may help the TA evaluate your solution if it only partially works.
  - The “blue cup” should be the largest region of the blue cup as determined by findContours(). You do not need to grab the bit of the stem that is severed when my hand occludes part of the blob. What I mean by this will become obvious when you have gottnen that far.
  - Repeat this for the green and red cups.
- Publish a rostopic for each cup. Each of these rostopics must be viewable in rviz and should use cv_bridge.
  - /color_filter/blue_cup
  - /color_filter/green_cup

  - /color_filter/red_cup

- Next, create a topic that has all 3 cups in one image. bitwise_or() can help you do this
  - /color_filter/cups
- Your solution should stream these topics so that they can be viewed as a video in rviz.

The Towers of Hanoi in PDDL

You can find FF at https://fai.cs.uni-saarland.de/hoffmann/ff/FF-v2.3.tgz

You will need to download and compile it. All of the tools should be available on machines in the BWI Lab (GDC 3.402) or and on CS lab machines, but if you download it onto your personal machine, you may also need to download flex and bison in order to compile it.

If you are unable to run ff, remember that you may need to put it into your $PATH.

Consider Figure 1.

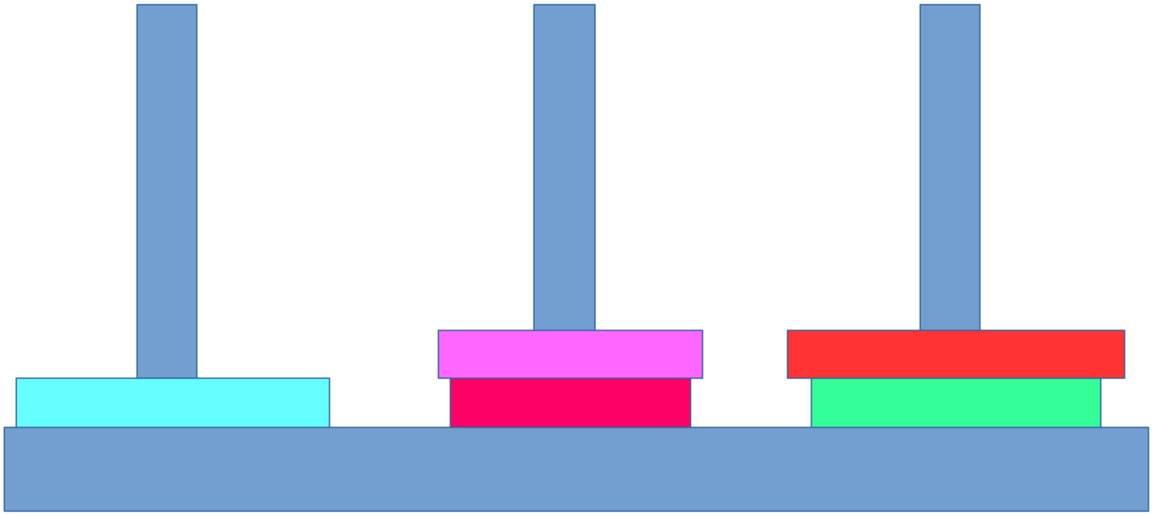

Figure 1: Your start state.

There are 3 posts. Each has rings on it. From bottom to top on each post. The first post has the second largest ring. The second post has the smallest ring, with the second smallest ring on top of it. The third post has the third largest ring, with the largest ring stacked on top of it.

The rules of this game are that you may only stack a ring on top of a larger ring. Your goal is to get all of the rings onto the same post, stacked from largest to smallest.

For the second part of your homework, write a PDDL domain and fact file encoding this Towers of Hanoi problem, such that FF can solve the Towers of Hanoi.