CSE 5120 Homework Assignment 2 – Fall 2021

Instructor: Dr. Kerstin Voigt

Total number of points: 20 (10+10)

This assignment will have you implement an A*-based program to solve the "Water Jug Problem", that has been introduced in the lecture video.

The "Water Jugs" Puzzle

You are given two jugs, one can hold 3 liters, the other can hold 4 liters.

There is an unlimited supply of water, and there is no penalty for wasting any. You may do the following with your water jugs:

1. You can fill any jug; this means you fill it to its maximal capacity.

2. You can dump any jug; this means you empty it completely.

3. You may pour water from one jug to the other; you will stop pouring when either the pouring jug has become empty, or the receiving jug has reached capacity.

Starting with two empty jugs, determine a sequence of operations (fill, dump, pour) that will result in an empty 3-liter jug and a 4-liter jug with exactly 2 liters of water.

Exercise 1: (10 pts) Before you focus on the "Water Jugs" puzzle, implement the generic A* algorithm as shown in the lecture slides. You may call this file generic_astar_search.py. Copy the implementation of the generic ASTAR_SEARCH function. A puzz8.py file is provided. Run your program for the 8-puzzle. It is still set to solve puzzE from puzz8.py. Try a few of the other 8-puzzles as well. Your code should run and produce solution paths.

Note: (1) Whether you use Python 2 or Python 3, change the line `open.sort(lambda x,y:…)`' with

        `open.sort(key = operator.attrgetter('theeval'))`

and add   `import operator`

to the import statements on the top of your file.

(2) All Python 2 print statements can be coverted to Python 3 quickly by wrapping a set of parenthesis ('( )') around the Python 2 string and '%...' if applicable. Pattern:

        print "aaaaa %d bbbb" % (…)   → print("aaaaa %d bbbb" % (…))

        print "aaaaa", → print("aaaaa", end="") suppressed newline
                                        character at end of line

Exercise 2: (10 pts) Slide 10 of the lecture slides shows you how to set up the call to generic ASTAR_SEARCH for the water jugs puzzle. You will need a new file, say, jugs.py which contains the goal, evaluation, successor and show functions for water jugs. Slide 10 already suggests that we represent two empty jugs (3-liter and 4-liter jugs) with a plain tuple [0,0]; [0,2] then is the goal state.

For this puzzle, it is difficult to come up with an evaluation that is informative and admissible (= always underestimates the true distance to the goal). ***While this seems meaningless, code an evaluation function that always returns 0 … at least, this function is guaranteed to be admissible and the solution found will be on the shortest path from the starting state*** (which is what we want for this puzzle). Should you be able to think of a truly brilliant alternative to evaluation function 0, as long as it is admissible, you may try it … but do not let this eat up too much of your time.

Friday, Oct 01

Submit by ~~Thursday, 9/28,~~ by 11:59 pm: Via Blackboard portal. (1) A copy of your file generic_astar_search.h, (2) a screenshot of the last moves of ASTAR_SEARCH solving an 8-puzzle, (3) a copy of your file jugs.py, and (4) a screenshot that shows all of the last moves of ASTAR_SEARCH solving the water jug puzzle. In case your program does not run, submit for items (2) or (4) a screenshot that shows the attempt to run the program and error messages.

Also, please adhere to the following naming rules when submitting your files.

File naming: For ease of identification of your submitted work, make sure that you adhere to the following **file naming convention:** for each file   XYZ.py, or screenshot that you submit, name the file

        Lastname_Firstname_####_XYZ.py(.<any other>)

where #### are the last four digits of your student id, and XYX is replaced with the name of yur file. **Your work may not be graded if you do not adhere to this naming convention.**