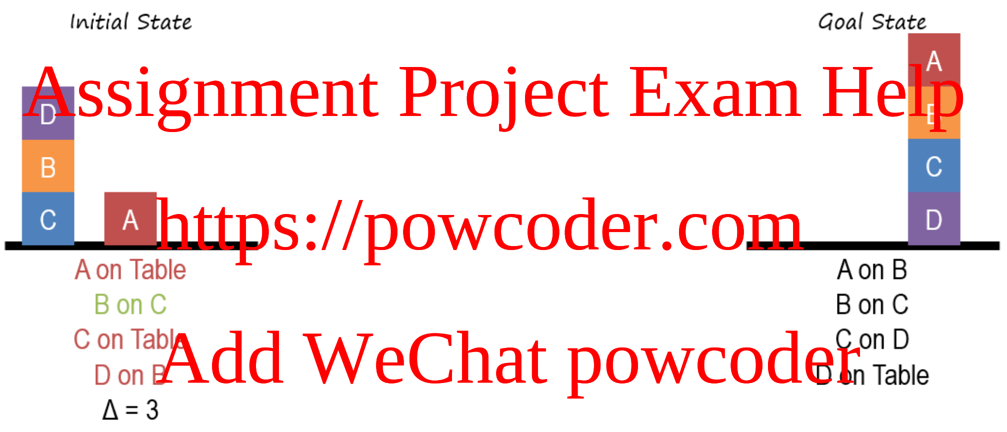
KBAI

Mini-Project 2: Block World (Spring 2021)

In this mini-project, you'll implement an agent that can solve Block World problems for an arbitrary initial arrangement of blocks. You will be given an initial arrangement of blocks and a goal arrangement of blocks, and return a list of moves that will transform the initial state into the goal state. You will submit the code for solving the problem to the Mini-Project 2 assignment in Gradescope. You will also submit a report describing your agent to Canvas. Your grade will be based on a combination of your report (50%) and your agent's performance (50%).

About the Project

In a Block World problem, you are given an original arrangement of blocks and a target arrangement of blocks, like this:



For us, blocks will be identified as single letters from A to Z.

Blocks may be moved one at a time. A block may not be moved if there is another block on top of it. Blocks may be placed either on the table or on top of another block. Your goal is to generate a list of moves that will turn the initial state into the goal state. In the example above, that could be: Move D to the table, move B to A, move C to D, move B to C, and move A to B.

There may be more than one sequence of moves that can accomplish the goal. If so, your goal is to generate the smallest number of moves that will turn the initial state into the goal state.

Your Agent

To write your agent, download the starter code below. Complete the `solve()` method, then upload it to Gradescope to test it against the autograder. Before the deadline, make sure to select your best performance in Gradescope as your submission to be graded.

Starter Code

Here is your starter code: BlockWorldAgent.zip. You may also access it from the course Github repository.

The starter code contains two files: BlockWorldAgent.py and main.py. You will write your agent in BlockWorldAgent.py. You may test your agent by running main.py. You will only submit BlockWorldAgent.py; you may modify main.py to test your agent with different inputs.

In BlockWorldAgent.py, your `solve()` method will have two parameters: the initial configuration of blocks, and the goal configuration of blocks. Configurations will be represented by lists of lists of characters, where each character represents a different block (e.g. "A" would be Block A). Within each list, each subsequent block is on top of the previous block in the list; the first block in the list is on the table. For example, this list would represent the configuration shown above: two stacks, one with D on B and B on C, and the other with just A:

```
[["C", "B", "D"], ["A"]]
```

There may be up to 26 blocks in a puzzle. You may assume that the goal configuration contains all the blocks and only the blocks present in the initial configuration.

Returning Your Solution

Your `solve()` method should return a list of moves that will convert the initial state into the goal state. Each move should be a 2-tuple. The first item in each 2-tuple should be what block is being moved, and the second item should be where it is being moved to—either the name of another block or "Table" if it is to be put into a new pile.

For example, imagine the following initial and target state:

Initial: `[["A", "B", "C"], ["D", "E"]]`
Goal: `[["A", "C"], ["D", "E", "B"]]`

Put in simple terms, the goal here is to move Block B from the middle of the pile on the left and onto the top of the pile on the right.

Given that, this sequence of moves would be an acceptable solution:

```
("C", "Table")
("B", "E")
("C", "A")
```

Submitting Your Solution

To submit your agent, go to the course in Canvas and click Gradescope on the left side. Then, select CS7637 if need be.

You will see an assignment named Mini-Project 2. Select this project, then drag your BlockWorldAgent.py file into the autograder. If you have multiple files, add them to a zip file and drag that zip file into the autograder.

When your submission is done running, you'll see your results.

How You Will Be Graded

Your agent will be run against 20 pairs of initial and goal configurations. 8 of these will be the same time every time your agent is tested; these are present in the original BlockWorldAgent.py file. The remaining 12 will be randomly generated, with up to 26 blocks each.

You can earn up to 40 points. You will earn 1 point for each of the 20 pairs of configurations you solve correctly (meaning that your solution does in fact transform the initial state into the goal state), and an additional point for each of the 20 configurations you solve *optimally* (in the minimum number of moves).

You may submit as many times as you want prior to the deadline. You **must** select which of your submissions you want to count for a grade prior to the deadline. Note that by default, Gradescope marks your last submission as your submission to be graded. We cannot automatically select your best submission. Your agent score is worth 50% of your overall mini-project grade.

Your Report

In addition to submitting your agent to Gradescope, you should also write up a short report describing your agent's design and performance. Your report may be up to 4 pages, and should answer the following questions:

- How does your agent work? Does it use Generate & Test? Means-Ends Analysis? Some other approach?
- How well does your agent perform? Does it struggle on any particular cases?
- How efficient is your agent? How does its performance change as the number of blocks?
- Does your agent do anything particularly clever to try to arrive at an answer more efficiently?
- How does your agent compare to a human? Does your agent solve the problem the same way you would?

You are encouraged but not required to include visuals and diagrams in your four page report. The primary goal of the report is to share with your classmates your approach, and to let you see your classmates' approaches. You may include code snippits if you think they are particularly novel, but please do not include the entirety of your code.

Submission Instructions

Complete your assignment using JDF, then save your submission as a PDF. Assignments should be submitted to the corresponding assignment submission page in Canvas. You should submit a **single** PDF for this assignment. This PDF will be ported over to Peer Feedback for peer review by your classmates. If your assignment involves things (like videos, working prototypes, etc.) that cannot be provided in PDF, you should provide them separately (through OneDrive, Google Drive, Dropbox, etc.) and submit a PDF that links to or otherwise describes how to access that material.

This is an individual assignment. All work you submit should be your own. Make sure to cite any sources you reference, and use quotes and in-line citations to mark any direct quotes.

Late work is not accepted without advanced agreement except in cases of medical or family emergencies. In the case of such an emergency, please contact the Dean of Students.

Grading Information

Your report is worth 50% of your mini-project grade. As such, your report will be graded on a 40-point scale coinciding with a rubric designed to mirror the questions above. Make sure to answer those questions; if any of the questions are irrelevant to the design of your agent, explain why.

Peer Review

After submission, your assignment will be ported to Peer Feedback for review by your classmates. Grading is *not* the primary function of this peer review process; the primary function is simply to give you the opportunity to read and comment on your classmates' ideas, and receive additional feedback on your own. All grades will come from the graders alone.

You receive 1.5 participation points for completing a peer review by the end of the day Thursday; 1.0 for completing a peer review by the end of the day Sunday; and 0.5 for completing it after Sunday but before the end of the semester. For more details, see the participation policy.