Q2: Writing your own (PDDL) planning problem

CS 695, Prof. Stein

Due: Wednesday September 27, 2022

Idea adapted from MIT 16.412

Here you will exercise some creativity! PDDL is used to solve quite a variety of scenarios including robotics, logistics, vehicle routing, and even genetics. In this problem, you should come up with your own domain and problem. I want you to think of a problem domain that is challenging enough that the solution isn't immediately obvious, and so you might consider implementing it in PDDL (even though you won't be doing much implementation for this problem). I will be looking for some creative solutions inspired by your own life or research. The problem doesn't need to be particularly complicated (I don't want you to spend too long on this), but try to think of a domain that has **at least three different operators.** Feel free to use the code I provided from P2 to get started and you can use the `downward-benchmarks` (code here) for inspiration.

Note: I don't want this question to feel like a ton of work. Short, to-the-point answers are preferred. Don't feel the need to write multiple paragraphs for each of these. Since there is only one question on this quiz, I'm giving you 50% credit as a starting point on this assignment, so that you don't feel too much pressure to be "perfect."

1. **Describe (in roughly 4–8 sentences) your problem setting.** What is your planning problem. Speak in a high level here and focus: I want to know that you can generally describe what you are trying to accomplish and why using PDDL is necessary—i.e., the solution should not be immediately obvious. For the block-stacking problem from class, I might start describing its domain by saying: *my problem setting is one in which a robot arm can pick up blocks and move them around.* Try to separate your *domain* from your *problem* if possible. Usually a *domain* can support multiple initial conditions and goal states. You

may use the description of the "Star Trek" example from P3 as a guide.

2. **Describe (in words) the different operators in your problem.** Next, describe (in words) what operators exist including their arguments/inputs, their preconditions, and their effects. If you need to describe other fluents or other quantities to help you in your description, feel free. You should describe at least 3 operators.

3. **Implement one of those operators in PDDL code and include it in your writeup. You do not need to implement your entire domain.** In fact, I will not even check that your PDDL code runs, so it's okay if there's some errors in it (like parentheses issues. What I am looking for here is that you can turn your english-language description of the operator into code. You can provide descriptions of each fluent you use if you would like, but as long as the name is fairly clear, you do not need to.

4. **Describe 2 *problems* in this domain.** Each problem is defined by an initial condition and a goal. They should be distinct from one another, so that your intiution for which actions are necessary or what the solutions will look like are different between the two. For at least one of these, describe why the solution is not immediately obvious. *Note:* as long as the goal cannot be reached in 3 or 5 actions, I would consider it *non-obvious*; it is sufficient to argue that the goal cannot be reached in fewer than 4 actions.