15-887 Homework 1

Assigned on: 9/21/16
Due date: 10/5/16 (beginning of class)

1 Lunar Lockout Game

Game Pieces

The lunar lockout game has several pieces. First, we have a 5x5 game board with a red square marked in the middle. Next, we have 5 helper spacecraft in various colors and then 1 red spacecraft. We are also given cards that specify initial setup position for some subset of spacecraft. On the back of each card is a solution.

(a) 5x5 Board (b) Helpers (c) Red (d) Puzzle 1 (e) Solution 1

Game Description

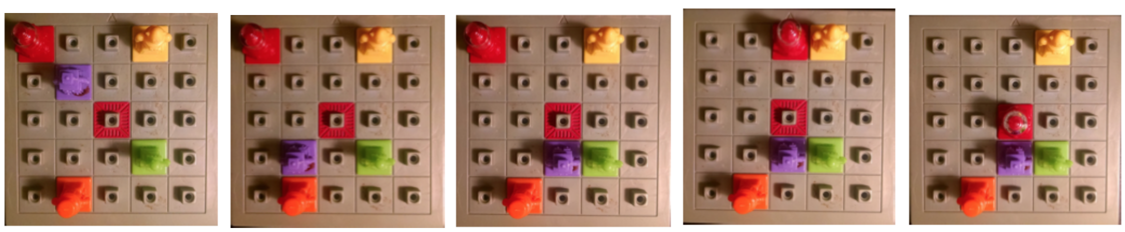
The goal of this game it to move the red spacecraft to the center red square. One can move any spacecraft but they are limited to moving up-down or left-right. **Whenever a spacecraft moves, it continues moving until it hits another spacecraft. Note that it is not legal for the spacecraft to move off the board.** Therefore, all spacecraft need to cooperate together to stay on the board and help the red spacecraft reach the goal.

Game Play

We present two simple beginner puzzles to show typical game play. Puzzle 1 is the beginner puzzle shown above. Starting from the initial setup, the red spacecraft moves up, left, down, then left to finally end up at the goal position. In this simple puzzle, only the red robot had to move. In the more complex puzzles, we also require other helper spacecraft to move. Puzzle 2 is an example requiring another helper robot to move as well. In this assignment, you will be defining a representation for this puzzle and also implementing a planner to solve them.

(a) Init (b) Move up (c) Move Left (d) Move Down (e) Move Left

Figure 2: Puzzle 1. The solution only requires moving the red spacecraft.



(a) Init (b) P - Move Down (c) P - Move Right (d) R - Move Right (e) R - Move Down

Figure 3: Puzzle 2. The solution requires first moving the purple spacecraft and then the red spacecraft.

a) [5 pts]

Define two very different ways to represent the lunar lockout world (objects, states, actions, domain axioms). Ensure there is a big difference in the set of possible states between both representations.

b) [5 pts]

Explain the number of possible states for both representations that you came up with. Do they also differ in the number of feasible states? Explain why each representation would be preferable to the other.

c) [45 pts]

Write a planner of your choice to solve this problem (Hint: a state-space search is likely the most simple). We should be able to run your code with something like the following command: *./lunar-lockout <domain file> <problem file>*. The *<domain file>* should define the lunar lockout world actions and predicates. If you wish, you can hardcode the domain representation into your planner but be sure to state this in your README. The *<problem file>* contains the objects, init state, and goal state. The output of your planner should be a sequence of actions that solves the given puzzles. Explain how we should interpret the output. It is OK to use a more compact representation.

Here are links to puzzles of increasing difficulty:

1. **Beginner** `http://www.cs.cmu.edu/~mmv/planning/homework/Homework1/cards.jpg`
2. **Intermediate** `http://www.cs.cmu.edu/~mmv/planning/homework/Homework1/cards1.jpg`

3. **Advanced** `http://www.cs.cmu.edu/~mmv/planning/homework/Homework1/cards2.jpg`
4. **Expert** `http://www.cs.cmu.edu/~mmv/planning/homework/Homework1/cards3.jpg`
5. **Expert 2** `http://www.cs.cmu.edu/~mmv/planning/homework/Homework1/cards4.jpg`

Please package your code in a zip file that includes <*problem file*> for at least puzzles 9, 18, 27, and 36. E-mail your code to mmv@cs.cmu.edu, maxim@cs.cmu.edu, and ruisilva@cmu.edu. The subject of your e-mail should be "15-887 Homework 1 - your andrew id" Although not required, we recommend using the PDDL language to represent the problem. That way you can evaluate your problem definition with existing planners like:

1. **FF planner**, similar to GraphPlan (`http://fai.cs.uni-saarland.de/hoffmann/ff.html`)
2. **VHPOP**, a partial order planner (`http://www.tempastic.org/vhpop/`)
3. **LPG**, a heuristic planner (`http://zeus.ing.unibs.it/lpg/`)

If you choose to go this route, feel free to include these planners and code used to evaluate your representation of the lunar lockout problem.

d) [5 pts]

With either your own planner or an existing one, run both representations of the lunar lockout problem that you came up with in Problem 1 for at least puzzles 9, 18, 27, and 36. Report the number of states needed to solve the problem for each representation. Discuss the results. Is one representation clearly better than the other? If not, explain why neither representation dominates.

2 Mean Ends Analysis

a) [5 pts]

Give a simple example where means-end analysis is not optimal. Show noteworthy steps of the search stack.

Contrived example

For the next subquestions consider the following contrived example domain

	op1	**op2**	**op3**
pre	—	g3	g4
add	g1	g4	g2
del	g2, g3	—	—

and the problem with **initial state**: g2, g3 and **goal state**: g1, g2.

b) [5 pts]

What is the solution plan for the given problem?

c) [5 pts]

Explain in detail, by showing the complete sequence of decisions and choices, why Prodigy with Means-Ends Analysis cannot find a solution plan.
Hint: You should carefully study the Prodigy algorithm. Read the paper available in the readings section. Studying the slides is probably not enough.

d) [5 pts]

Show how GraphPlan would solve this example.

e) [5 pts]

Explain how Prodigy could be extended to avoid incompleteness in domains like this one.

3 GraphPlan

a) [5 pts]

GraphPlan can backtrack during its second phase while searching backwards from the goals through the planning graph. Show with a simple example, why GraphPlan needs to backtrack in its backwards search.

b) [5 pts]

If GraphPlan cannot find a solution during the backwards search, does this mean that the problem is not solvable? If so, explain why. If not, explain what GraphPlan can do to find a solution in such cases.

c) [5 pts]

It is possible for GraphPlan to terminate after finding an n time step plan of k operators, while there actually exists an $n+1$ time step plan with less than k operators in the planning graph. Show a concrete example of this and explain why, in general, this can occur.