

CS531 Programming Assignment 1

Vacuum-Cleaning Agents

Due: Jan 20'th, 8:30 AM

In this assignment you will design and implement 3 different vacuum-cleaning agents. You can do this and other programming assignments in teams of 2 students. You can use any programming language for this assignment.

The vacuum cleaner environment consists of $n \times m$ empty rectangular room, where each square has a $p\%$ chance of containing dirt. There are 3 percepts: a wall sensor = 1 if the machine has a wall right in the front and 0 otherwise, a dirt sensor = 1 if the square contains dirt, and a home sensor = 1 if the agent is home (the starting location). Five actions are available: go forward, turn right by 90 degrees, turn left by 90 degrees, suck up dirt, and turn off. The agent always starts in the bottom leftmost corner oriented upwards. It is evaluated by the total number of clean cells as a function of the number of actions taken. (Plot the number of clean cells vs. the number of actions taken). Design and implement three different agents and experiment with $m=n=10$ and $p=100\%$. The three agents are listed as follows. In each case, the agent program should be organized as a list of if-then rules. The first rule whose condition evaluates to TRUE will "fire," which means the corresponding action is executed. If the agent program is deterministic then there is only one possible action selected at each step. If it is stochastic, then there are several actions, and each action will be chosen with some probability. For example, the rule "if wall then turn-left 0.5 turn-right 0.5" will choose turn-left and turn-right with equal probability if the wall sensor is ON.

1. A simple memoryless deterministic reflex agent.
2. A randomized reflex agent that can choose actions randomly based on sensor readings.
3. A deterministic model-based reflex agent with a small amount (2 to 3 bits) of memory that represents the "state." When executing each action, the agent simultaneously updates the state by setting or resetting these bits according to how you specify them. The actions can be based on its current state bits as well as the current percepts.

Important Note: Note that the agents are not allowed to take multiple actions in sequence on seeing a percept. For example you cannot write rules that say "if facing wall then turn right and go forward". Similarly the agents cannot base their actions on percept histories. They can only base it on the current percept plus the state information if any.

For the random agent, repeat the experiment 50 times and plot the average performance over the 50 runs. For other agents, one run each would do. Write a brief report (3-5 pages). Your report should look like a research paper with abstract, introduction (a summary of the goals and results), formal description of the agents in some rule-like notation, experimental setup, results with 3 plots, and the discussion. You should include answers to the following questions in your papers. Please feel free to add any other important observations as well.

1. Describe the idea behind the design of each of your agents. Use diagrams and english as appropriate.
2. Describe all your agents as if-then rules.
3. What is the best possible performance achievable by any memory-less reflex agent in this domain?

What prevents a memory-less reflex agent from doing very well in this task?

4. How well does the random agent perform? Do you think that this is the best possible performance achievable by any random agent? Why or why not? Give a table showing the number of actions it took to clean 90% of the room for each trial. What is the average of these numbers for the best 45 trials? What are the costs and benefits of randomness in agent design?
5. How does the memory-based deterministic agent perform compared to the random agent? Was it able to completely clean the room? Was it able to shut itself off after it is done? If it did, how many actions did it take to do this? Can the agent be improved any further with more memory than you used? Why or why not?
6. What did you learn from this experiment? Were you surprised by anything?

Note: You are not required to build any graphical user interface for this program.

Submit the source code using the [ENGR assignment submission system](#). Login and click the "Submit an Assignment" button. Submit the report (hard copy) in the class on the due date.
