EECS 492: Introduction to AI
Homework 1 (100 pts)

Problem 1: Characterizing an Agent's Environment [28 points]

In class we talked about specifying a task environment via the PEAS description. The PEAS description contains the following specifications: Performance Measure, Environment, Actuators, and Sensors.

In addition, environments can be described in terms of the following properties:

- Fully/Partially observable
- Deterministic/Stochastic
- Episodic/Sequential
- Static/Dynamic
- Discrete/Continuous
- Single-agent/Multi-agent

Describe the following agents using (A) the PEAS description and **also** (B) a description of the task environment. Justify your answers, briefly (no more than 1 or 2 sentences per term). Each question is worth 4 points (2 for PEAS and 2 for task environment).

1. Crossword Puzzle Solver

P. How much of the puzzle is filled in with letters that spell correct English words across rows and columns. The more the puzzle is filled in this way, the higher the performance.

E. A partial 2d grid of adjacent squares, with numbers labeling the rows and columns, and a list of written clues associated with each number.

A. Whatever allows the agent to write letters - a bit of computer code for a computer crossword solver, or a pencil for a real-world robot solving a crossword.

S. Whatever allows the agent to read the clues and the letters, and the structure of the environment - some computer code if the agent is a computer program, or cameras otherwise.

Fully vs. Partially Observable: Fully observable
Deterministic vs. Stochastic: Deterministic
Episodic vs. Sequential: Sequential
Static vs. Dynamic: Static
Discrete vs. Continuous: Discrete
Single-agent vs. Multi-agent: Single-agent

2. Roomba (https://en.wikipedia.org/wiki/Roomba)

P. The cleaner the floor, the better the performance
E. A house

A. Wheels and a motor to drive them
S. IR, Odometry
Fully vs. Partially Observable: Partially observable
Deterministic vs. Stochastic: Mostly deterministic
Episodic vs. Sequential: Episodic
Static vs. Dynamic: dynamic
Discrete vs. Continuous: Continuous
Single-agent vs. Multi-agent: Single-agent

3. Netflix Automatic Recommender System

P. The more accurate recommendations made, the better the performance (i.e., the more people who watch the recommended movies and give them about the predicted rating, the better)

E. A web interface, with data from users coming in in real time.

A. Whatever code allows the agent to show users new movies with predicted ratings.

S. Code to parse the incoming user data.

Fully vs. Partially Observable: Partially observable: not all users rate the movies they have seen

Deterministic vs. Stochastic: Stochastic

Episodic vs. Sequential: Sequential - performance can improve continuously over time.

Static vs. Dynamic: Dynamic: user preferences can change over time, as can the movies that they watch

Discrete vs. Continuous: Discrete, because ratings are discrete and movies as a whole are discrete

Single-agent vs. Multi-agent: Single-agent, effectively. Humans are part of the environment.

4. Home Alarm

P. The larger the ratio of true positives to false positives, the better

E. A house

A. The speakers that sound the alarm; the interactive screens for the owners to turn the alarm on and off

S. The buttons next to the interactive screens; the lasers, vibration sensors, and other things that detect burglary.

Fully vs. Partially Observable: Partially observable - not every part of the house can be directly seen, feasibly.

Deterministic vs. Stochastic: Stochastic - sometimes things set off the alarm that aren't supposed to, sometimes things aren't detected that should be detected

Episodic vs. Sequential: Episodic: different days are generally independent of each other

Static vs. Dynamic: Dynamic: people and animals can come and go continuously

Discrete vs. Continuous: The environment is continuous, though it could be argued that the sensors are discrete

Single-agent vs. Multi-agent: Single-agent

5. Boston Dynamics BigDog (https://www.youtube.com/watch?v=cNZPRsrwumQ)

P. The longer and farther it can move through an environment while staying upright and not get lost or fall over, the better.

E. Real-world terrain: forests, hills, fields, rocky shores, etc.

A. Robotic legs that propel the robot forward and keep it upright

S. A variety of internal and external sensors, including pressure and position sensors on its legs, oil and temperature sensors, and others.

Fully vs. Partially Observable: Partially observable: objects occlude each other in real-world environments

Deterministic vs. Stochastic: Stochastic: the environment may change unpredictably

Episodic vs. Sequential: Sequential, for one journey - future performance in terms of, say, distance traveled, is affected by what was done in the past. Could also argue episodic, where each particular use of BigDog does not affect the others.

Static vs. Dynamic: Dynamic - the environment may change in temperature, lighting, or due to actions of animals and humans.

Discrete vs. Continuous: Continuous

Single-agent vs. Multi-agent: Single-agent

6. Siri (https://www.apple.com/ios/siri/)

P. More productive interactions with users: less misunderstanding; performance the task the users asks for more quickly and more accurately

E. The internet, the phone application environment

A. Speakers to produce voice commands, code to interact with other applications

S. Microphones to listen to the user, code to sense typed in user commands

Fully vs. Partially Observable: Partially observable - there is too much data in existence for Siri to see it all at once

Deterministic vs. Stochastic: Stochastic: certain actions may have different effects, especially when performing internet searches

Episodic vs. Sequential: Episodic: Siri interacts with people in discrete events

Static vs. Dynamic: Static on the timescale that Siri operates, though could argue dynamic

Discrete vs. Continuous: Mostly discrete, though could argue that data from microphone is continuous

Single-agent vs. Multi-agent: Single-agent

7. Paro Therapeutic Robot (http://www.parorobots.com/)

P. How much patients enjoy interacting with the robot, and improve in well-being because of it.

E. Nursing homes and hospitals

A. Electric motors to move its body around

S. Light, posture, tactile, auditory, and temperature sensors

Fully vs. Partially Observable: Partially observable: the robot does not have enough sensors to fully see its environment

Deterministic vs. Stochastic: Stochastic - it cannot perfectly predict how people will react to it, or what to do next

Episodic vs. Sequential: Patient encounters are episodic

Static vs. Dynamic: Dynamic - people can change position / attitude while the robot is operating

Discrete vs. Continuous: Tactile and light data is continuous

Single-agent vs. Multi-agent: Single-agent

Problem 2, Parts 1-4

See code file.

Problem 2, Part 5

For Problem 2, include answers to the following questions in your pdf. Questions are worth 4 pts each, unless otherwise specified.

1. Write a PEAS description for FA-bot's environment in Part 1.

Performance: Whether all the weeds are weeded, and squares that need watering watered, before the agent runs out of power, and also whether or not power and water are conserved as much as possible.

Environment: The farmworld, with barriers, weed indicators, and watering indicators, arranged in a 2d grid.

Actuators: Whatever allows the agent to perform the actions in the spec. Wheels to move around, and arms for weeding and watering, for example.

Sensors: Whatever allows the agent to gather the sense data in the spec: cameras, soil sensors, etc.

2. What is the definition of rational behavior in FA-bot's environment?

Rational behavior here means weeding and watering all necessary squares while conserving and watering as much as possible, given all past percepts and available information to the agent.

3. Can the simple reflex agent from Part 1 be perfectly rational for this environment? What about the agent from Part 2? Why or why not?

Justification for no in both cases: the agent from part 1 cannot be perfectly rational, since it ignores information such as past data, leading to suboptimal performance. Similarly, the agent from part 2 must ignore information about where it has already been, since it can only maintain limited state.

Justification for yes in both cases: both agents are perfectly rational given *the constraints that are placed upon them - they do as well as possible among the set of similarly constrained agents.*

4. For **each** of parts 1, 2, 3, and 4, fill in a table like the following that contains the outcomes of each agent running on **each** provided environment. (For the randomized agent, there is, of course, no single correct answer. Provide averages over 5 runs, with percent of times succeeded in the "succeeded" column).

Simple reflex:

Env. #	Succeeded?	# Moves	Water	Power	# Left to water	# Left to weed
1	Yes	27	1	73	0	0
2	No	25	5	0	1	1
3	Yes	30	3	70	0	0
4	Yes	42	5	58	0	0
5	No	100	7	0	3	1

State reflex:

Env. #	Succeeded?	# Moves	Water	Power	# Left to water	# Left to weed
1	Yes	23	1	77	0	0
2	Yes	7	4	18	0	0
3	Yes	27	3	73	0	0
4	Yes	37	5	63	0	0
5	Yes	47	4	53	0	0

Random Reflex:

Env. #	Succeeded?	# Moves	Water	Power	# Left to water	# Left to weed
1	0.6	82	1.2	16	0.2	0.4
2	0.8	11.8	1.05	13.2	0.2	0.2
3	0.8	67.4	1.07	32.6	0.2	0.2
4	0.4	95.2	6.2	4.8	1.2	0.4
5	0.2	97.6	5.4	2.4	1.4	1

Better reflex:

Env. #	Succeeded?	# Moves	Water	Power	# Left to water	# Left to weed
1	Yes	25	1	75	0	0
2	Yes	22	4	3	0	0
3	Yes	30	3	70	0	0
4	Yes	42	5	58	0	0
5	No	100	7	0	3	1

5. Which agent performed the best overall? Why? Does your randomized simple reflex agent from Part 3 outperform your deterministic agent from Part 1 in any environments, and why? List the ones that it does, if any.

The reflex agent from Part 2, which does search on its environment beforehand, performs the best, because it does not waste any power on unnecessary movements, while sticking to the best strategy in terms of sensing.

The random agent frequently outperforms the simple reflex agent in environment 2, because it can (serendipitously) find the only square that needs weeding/watering faster.

6. (7 pts) In which environment(s) did your agent from Part 4 outperform the agent from Part 1? Describe how you structured the better reflex agent, and why it performs better in some cases and no worse in others. (Or, if you did not succeed in making such an agent, describe what you were trying to do).

The better reflex agent outperformed the simple reflex agent in environments 1 and 2. In both environments, it conserved more power, and in environment 2, it actually conserved enough power to succeed where the simple reflex agent failed. The reason that this agent conserves more power is that it does not waste energy sensing squares that it has already visited. This is why there was no improvement in environments 3-5: the simple reflex agent did not repeat any part of its path in those cases. We designed the better reflex agent to contain state variables that track where the agent has been. Since the only interaction with the environment is through a set of basic percepts, the agent must define its own coordinate system relative to its initial position and direction. It does not need to know its absolute initial position and direction to uniquely identify every square that it visits - every square is seen as relative to the initial state, whatever that was. Whenever the agent successfully moves, the coordinate axis is either incremented or decremented, and when it hits a barrier, the coordinate axis to be updated and the increment/decrement indicator are flipped. Squares are added to a "visited" list after they have been sensed, and possibly watered and weeded. After every move, the agent makes sure that the current square is not in the visited list before sensing. Otherwise, it moves on.

Resource constraints lead to trade-offs between different strategies that agents could implement. Consider the following simplified scenario: an agent must move along a single row of grid squares, which either do or don't contain weeds with about 50% probability (e.g., the grid looks like F T T F T …). The agent can move forward, sense for weeds, and weed (like our agents above, the weed action can be performed whether or not there is actually a weed). The agent wants to weed all squares while using as little power as possible.

7. Describe this agent's optimal strategy assuming it costs 1 unit of power to sense for weeds and 5 units of power to weed.

The optimal strategy is to sense the environment at every new grid square, and only weed if it senses a weed. In expectation this will save the most power, since weeding it expensive compared to sensing.

8. How does this agent's strategy change if it costs 5 units of power to sense for weeds and 1 unit of power to weed?

   The agent's optimal strategy is to forgo sensing and just blindly weed every square, since this will now save it the most power in expectation. Weeding is now cheap enough compared to sensing that sensing is simply a waste of power.

The internal logic of our agents above was greatly simplified by *environmental determinism*. We knew that if the agent attempted a “weed” action, for example, the action would succeed with certainty. But, what happens when certainty is no longer guaranteed? Assume that we are working with the state-based reflex FA-bot that you implemented from Part 2.

9. Now our FA-bot is on the fritz. We have been meaning to replace the agent, but, we haven’t. As a result, the “weed” action only succeeds in removing weeds about 75% of the time! How would the performance of the state-based agent from Part 2 be affected by this indeterminism?
   As-is, the agent would not be guaranteed of accomplishing its goal of having all the weeds pulled out. Indeed, after it finished its stored path, it would not necessarily know what to do next, and may waste its power doing nothing.

10. Describe how you would you modify that agent to handle this sort of indeterminism.
    Add a loop that senses the square again after every attempt at weeding, until the action has been accomplished.