EECS 649: PROBLEM SET #2

Reading:
- R&N 2.1-4, Braitenberg's *Vehicles*, pp. 1-42 (linked)

Total Points: 100

Format:
- Use standard sheets of paper (8.5 by 11 inches) for your scans
- Perform all work neatly. When asked to write a paragraph, it should be typed (e.g., using a computer and LaTeX or Word).

Problem 2.1 [20 points]
For each of the following, give a PEAS description of the task environment -- in a grid as in Figure 2.5 of R&N -- and characterize it in terms of the properties described in Section 2.3.2 of R&N, putting your answers to the **that** part in a grid as in Figure 2.6 of R&N.
- Exploring the subsurface oceans of Titan
- Shopping for used AI books on the Internet

Problem 2.2 [10 points]
The discrete photovore we talked about in class had a one-dimensional percept with two values: {NoLight, Light}. Thus, using the notation in the middle of page 48 of R&N it has |P|=2 (that is, the total number of percepts is two).

It also had two actions A={Stay, Move}, so |A|=2. Thus, there are a total of 2^2 stimulus/response (S/R) agent programs or **agent functions** (also known as look-up tables or LUTs):

```
NoLight | Stay
Light   | Stay

NoLight | Stay
Light   | Move

NoLight | Move
Light   | Stay

NoLight | Move
Light   | Move
```

How many S/R agents are there in general? Write your answer in terms of the number of percepts and actions, |P| and |A|. Hint: You may wish to test your answer in special cases, such as |A| = 1 or |A| = 2 while |P| = n.

Use your answer to compute the number of S/R agents for the vacuum-cleaner agent pictured in Figure 2.2 (and described on page 37) of R&N.

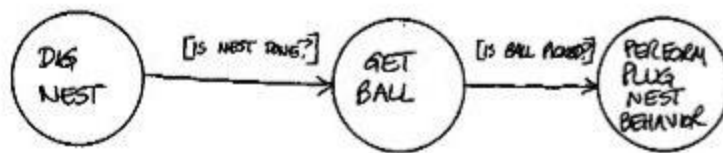*Hint: Think of the number of functions from the domain to the range.*

Problem 2.3 [10 points]
Draw a state machine capturing the behavior of the sphex wasp described on page 41 of R&N.

SAMPLE Problem Like Problem 2.3
Draw a state machine capturing the behavior of the **dung beetle** described on page 41 of R&N.

Sample Answer:

Problem 2.4 [10 points]
Design (the stimulus response curves and connections for) a vehicle that circles a source at a fixed distance, as does the middle vehicle in Braitenberg's *Vehicles,* Figure 7 (page 18).

Problem 2.5 [10 points]
Recall the equations we wrote in class for Figure 10b on page 23 of Braitenberg's *Vehicles*. Do the same for the circuit in Figure 10a, calling the input s, and the outputs of the successive threshold units (left to right) v, w, x, y, and z.

You may do this by hand or write a computer program to figure these out. Your choice. Whatever choice, make sure the answer is organized in a well-labeled table as we did in class.

Problem 2.6 [10 points]
   a. Draw the memory circuit described for Vehicle 5 (at the top of page 24 of *Vehicles*).
   b. Draw one that will remember if it has seen more than one red light.

Note: we will use the discrete-time model of Problem 2.5 above. Thus, "2 lights" means a red light was seen at two different integer times (which might or might not be contiguous).

Thus, you should draw a total of TWO circuits:

(a) the indicated circuit from the book, which latches (and continues to ring) if the vehicle has seen a light in <u>at least one</u> previous time instance
(b) one of your own design that latches/rings if the vehicle has seen a light in <u>at least two</u> different time instances

As suggested above, to help make your designs more specific, include the buzzer/bell (like Braitenberg's example did), whose ringing signifies the condition in question.

Problem 2.7 [30 points]
Implement a simple reflex agent for the vacuum environment in Figure 2.2. Run the environment with this agent for all possible initial dirt configurations and agent locations. Record the performance score for each configuration and the overall average score.

Note: this requires building a simulator for the vacuum environment in Figure 2.2 and described on page 37. You may write it yourself or use/start with code in R&N's code repository: https://github.com/aimacode/aima-python/blob/master/vacuum_world.ipynb
If you write your own simulator, you do **not** have to make a modular implementation in terms of size and shape, only initial configurations of dirt and robot. You **only** have to simulate the specific size and shape of the environment depicted in Figure 2.2.

NOTE: you may assume that <u>once clean, the rooms stay clean</u>.

There is **no need** to turn in any code. Just turn in a short explanation of how your program is structured. Cite any repository code used and note any significant modifications.

You must report the S/R table used and the performance score for each configuration and its overall average score.

Here are some more details that may help you out.

Overview. You need to write a simulator for a stimulus-response agent (of your design) acting in the 1x2 world. You need to compute its performance measure (of your design) for each of the 8 possible worlds.

The simulation. Your simulator need only work for a stimulus-response agent acting in the 1x2 world. Your simulator should be able to take an arbitrary environment (of the eight possible) as the start location. In that case, you would have to run it 8 times to gather all 8 performance measures for your agent. Alternatively, you could easily loop over all environments, simulating your agent in each one to obtain its performance measure in that environment.

The S/R agent. You need to make an STIMULUS-RESPONSE agent, not an arbitrary agent. For example, my S-R agent (yours can vary) is the following:

> *A, Clean --> Left*
> *A, Dirty --> Left*
> *B, Clean --> Left*
> *B, Dirty --> Left*

If this agent started in room B, with both squares initially dirty, it would move to room A after the first step and then (trying to move Left from there on each step, but running into the wall each time) stay there for as many steps as the simulation lasted.

The performance measure. This is not stated and is up to you. An example of one that makes sense to me for judging/comparing different S/R agents in this environment is as follows: simulate the agent for 10 steps, penalizing it 1 point for each square that is dirty in each time step. So, for the initial state and agent mentioned in the previous point, the performance would be -2 per time step (both rooms are dirty for each time step).

Write-up. Your writeup definitely needs to detail the S/R agent you used as well as the performance measure you used. You also must report the the performace measure on each of the environemts. Finally, it would be appropriate to BRIEFLY explain a little about how your simulator code works (data structures, looping, etc.)