COMP-5361: Discrete Structures and Formal Languages
Programming Assignment 2

Due: April, 1st, 2019
11:59PM

---

1 Introduction

In this assignment we will use the `FrozenLake-v0` environment to exploer our knowledge of `functions`! The environment and its backstory are presented below:

> Winter is here. You and your friends were tossing around a frisbee at the park when you made a wild throw that left the frisbee out in the middle of the lake. The water is mostly frozen, but there are a few holes where the ice has melted. If you step into one of those holes, you'll fall into the freezing water. At this time, there's an international frisbee shortage, so it's absolutely imperative that you navigate across the lake and retrieve the disc. However, the ice is slippery, so you won't always move in the direction you intend.The episode ends when you reach the goal or fall in a hole.

You are allowed to use any library you want for this assignment. You are required to use `matplotlib` for the plots and use of `numpy` would really speed up your progress. Before starting any experiments, please read the documentations and demonstrations presented by `Gym OpenAI` on the use of their environments where `FrozenLake-v0` is an example we chose to work with in this assignment.

2 Manual Inspection

Inspect the environment of the agent and come up with a fixed set of action that would lead you to the frisbee, as soon as possible i.e. smallest number of actions needed to get to the frisbee. What are the actions you need to take?

Since the ice is slippery, which means you will not always end up where you intended to go, if you follow the set of actions you provided earlier, you may not get to the frisbee! Run this experiment for 1000 times and report the percentage of the times you successfully end up with the frisbee and the times you would fail. Also show this comparison using a bar graph, where:

- The percentage of success has color red and the percentage of failure has color blue.
- The vertical and horizontal axis of the graph are labeled properly.
- The graph has a proper title.
- A proper legend is provided for the two outcomes.

3 Define a Function

In this part we aim to replace the fixed list of actions with a dynamic function. You need to define a simple function that would tell you the action you need to take based on your position.

$$f(\text{position}) = \text{best action } \forall \text{position} \in \text{Grid}$$

Explain the logic behind your function and how it works in details. Explain what kind of function is it and why. (injective, surjective, etc.)

Note that at the beginning you are located at the top left corner of this world and you aim to go to the bottom right corner and your function should be based on this fact. Now, do the same analysis we did in the previous section, i.e. run the experiment with your function for 1000 times, report on the percentage of the success and failure, and have a bar graph for the results with the same requirements as before. Compare your result to ones found in the previous section and explain the differences.

4 Extended List

In this part you are required to make the same type of list you made in Section 2 but this time for all other locations. In Section 2 the list you made was the smallest list of actions that would guide you from the top left corner to the Frisbee, now we want to have the same of type of list for each of the other locations (i.e. you would start the search for your frisbee from those locations). Let this list (function) be called h, then you need to define:

$$h(\text{position}) = \text{one minimal list of actions to the frisbee } \forall \text{position} \in \text{Grid}$$

5 Piecewise Function

In this part we will try to define another function for navigation in this environment that takes advantage of previous approaches (seen in Section 3, 4). We will define a Piecewise function k as follows:

$$k(\text{position}, \alpha) = \begin{cases} f(\text{position}) & \text{with probability } \alpha \\ h(\text{position}) & \text{with probability } 1 - \alpha \end{cases}$$

Where $\alpha \in [0, 1]$. Measure the performance of k for $\alpha \in \{0, 0.1, 0.2, \ldots, 1\}$ over 1000 runs show the plot of performance vs. values of α. What is the best value of α according to your plot? Explain.

6 What to Submit

Using the EAS system, submit a `.zip` file containg:

1. The ipython notebook as `.ipynb`.
2. A PDF of the ipython notebook.

The format to be followed for submission is:

```
yourstudentID_assignment1_yourname.zip
```

Make sure you use the `text cells` to explain what each of your functions, classes, or pieces of code do (e.g. input, output, functionality etc.). You will be graded based on:

- Correctness of your solutions.
- Correctness of your program.
- Organization and readability of your code.