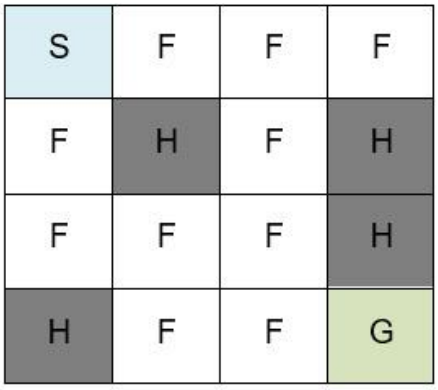
Assignment:2

Implementing Policy and Value Iteration on the FrozenLake-v1 Environment

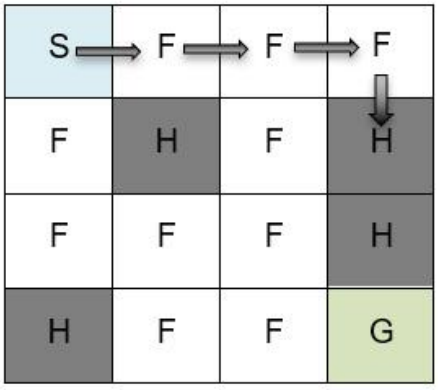
Marks 80

Frozen Lake is another simple game found in the OpenAI framework. This is a classic game where you can do sampling and simulations for Monte Carlo reinforcement learning. We have already described and used the Frozen Lake environment.

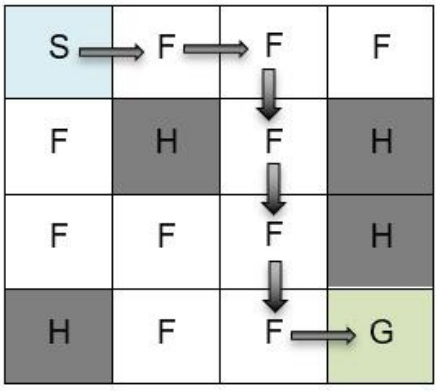
We have a 4x4 grid of cells, which is the entire frozen lake. It contains 16 cells (a 4x4 grid). The cells are marked as S – Start, F – Frozen, H – Hole, and G – Goal. The player needs to move from the Start cell, S, to the Goal cell, along with the Frozen areas (F cells), without falling into Holes (H cells). The following figure visually presents the aforementioned information:



A) Frozen Lake Grid



B) Failed Game ending by agent falling into a hole



C) Winning the Game by agent reaching the goal

Here are some basic details of the game:

- The aim of the game: The aim of the game is to move from the Start (cell S) to the Goal (cell G).
- States = 16
- Actions = 4
- Total state-action pairs = 64
- Strategy: The player should move along the Frozen cells (cells F) without falling into the Holes in the lake (cells H). Reaching the Goal (cell G) or falling into any Hole (cells H) ends the game.
- Actions: The actions that can be performed in any cell are left, down, right, and up.
- Players: It is a single-player game.
- Rewards: The reward for moving into a Frozen cell is 0 (cells F), +1 for reaching the Goal (cell G), and 0 for falling into a Hole (cells H).
- Configuration: You can configure the frozen lake to be slippery or non-slippery. If the frozen lake is slippery, then the intended action and the actual action can vary, so if someone wants to move left, they might end up moving right or down or up. If the frozen lake is non-slippery, the intended action and the actual action are always aligned. The grid has 16 possible cells where the agent can be at any point in time. The agent can take 4 possible actions in each of these cells. So, there are 64 possibilities in the game, whose likelihood is updated based on the learning. In the next activity, we will learn more about the Frozen Lake game, and understand the various steps and actions.

Note:

For more information on the FrozenLake-v1 environment, please refer to the following link:

https://gym.openai.com/envs/FrozenLake-v1/

Let's now implement the policy and value iteration techniques to solve the problem and retrieve the frisbee.

In this activity, we will solve FrozenLake-v1 using policy and value iteration. The goal of the activity is to define a safe path through the frozen lake and retrieve the frisbee. The episode ends when the goal is achieved or when the agent falls into the hole. The following steps will help you complete the activity:

In this activity, we will solve FrozenLake-v1 using policy and value iteration. The goal of the activity is to define a safe path through the frozen lake and retrieve the frisbee. The episode ends when the goal is achieved or when the agent falls into the hole. The following steps will help you complete the activity:

1. Import the required libraries: numpy and gym.
2. Initialize the environment and reset the current one. Set is_slippery=False in the initializer. Show the size of the action space and the number of possible states.
3. Perform policy evaluation iterations until the smallest change is less than smallest_change.
4. Perform policy improvement using the Bellman optimality equation.
5. Find the most optimal policy for the FrozenLake-v1 environment using policy iteration.

6. Perform a test pass on the FrozenLake-v1 environment.
7. Take steps through the FrozenLake-v1 environment randomly.
8. Perform value iteration to find the most optimal policy for the FrozenLake-v1 environment.

Note that the aim here is to make sure the reward value for each action should be one (or close to one) to ensure maximum rewards.