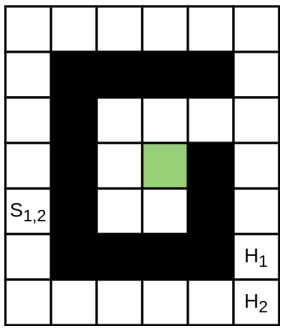
Homework 1 - A game of Hide and Seek and The Observant Maze Runner

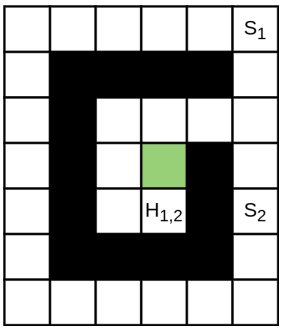
1 Part 1

The goal of this first assignment is to experiment with **model-free** reinforcement learning approaches to learn how to play hide and seek. The environment that we are going to use is a grid world that contains two pairs of agents, the hiders and the seekers. The state of the environment is fully determined by the positions of the seekers, the position of the hiders and the auxiliary movable box that helps the hiders to win the game (as it will shortly be explained).

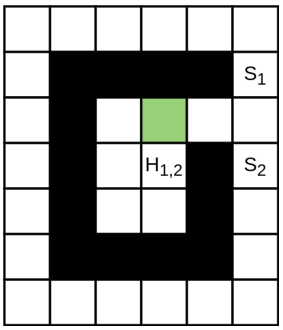
(a) Initial state



(b) Behind the box.



(c) Box pushed.



(d) Move away

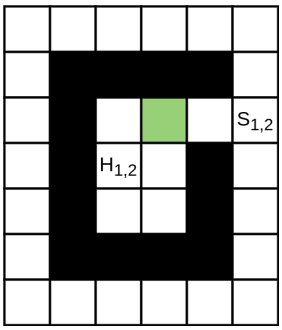


Figure 1

The initial state is deterministic and it is depicted the leftmost view from Figure 1. In this world, the seekers follow a predefined deterministic strategy to capture the hiders, by flanking them and moving toward the center. The environment is specifically designed to offer a safe place in the center of the map for hider agents. In order to win the game, the hiders need to develop an **intelligent** behavior, namely to push the movable box to block the entrance, as the block can not be moved by the seeker agents. Unfortunately the box is too heavy for a single agent, thus both hiders need to **cooperate** to block the entrance and win the game. First, they need to go in the south cell of the box and then move north at the same time. Once they have blocked the entrance, the hiders need to move away from the entrance's vicinity since a seeker can spot them from a distance of one cell (this includes diagonal).

Tasks

1. Implement an off-policy (Q-learning) and on-policy(Sarsa) TD control with ϵ-greedy exploration.
2. Experiment with the following variants of explorations strategies: constant $\epsilon = 0.1$, linear decay and exponential decay (from 1.0 to 0.1).
3. Experiment with the following learning rates(α): 0.1, 0.5.
4. Experiment with the following decay rates(γ): 0.9, 0.99, 1.00. Which one should be more appropriate in our setup?

Result Presentation

- Plot learning diagrams for each task and discuss the results.
- For tasks 2 - 4, vary only the parameter indicated in the task, keeping the rest at the **best value** you found in your experiments.
- Overlay your results on the same chart, such that they become easily comparable (e.g. for task 2 overlay learning curves for each ϵ value with a different color)
- Submit your results in a PDF file which will be used when presenting the homework.

2 Part 2

The goal of this second assignment is to observe the requirement of using POMDP models to solve simple games where exact state information is unavailable.

The game consists of a simple maze environment as depicted in Figure 2. The state space consists of 5 grid cells, whereby the initial belief of the agent is

$(0.5, 0, 0, 0, 0.5)$, that is it has a 50% / 50% chance of starting in one of the two states *SL* and *SR*. The action space consists of only two actions (*left* and *right*) and the actions have deterministic effects. The agent receives a reward of -1 for each move and a reward of 0 when reaching the goal state.

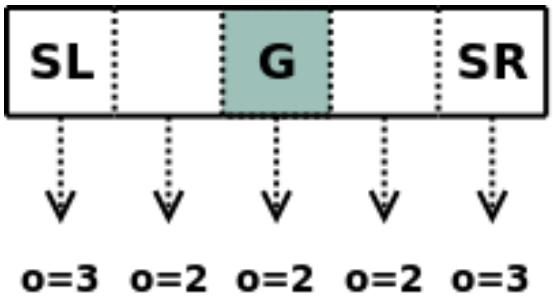


Figure 2: The agent can start in two initial positions, **SL** and **SR** and needs to find his way to the goal state **G**. The only observations perceived by the agent are the **number of walls** surrounding its current cell (i.e. $o = 3$ for SL and SR and $o = 2$ for the rest

Tasks

1. Model the game as a POMDP. Write down the state-action transition probabilities $p(s_j|a, s_i)$, the state-action reward matrix $r(s, a)$, the conditional observation probabilities $p(o|s)$, as well as the initial probability distribution over world states $b(s)$.

2. Use the **Point-based Value Iteration** algorithm presented in Lab 4 to implement a solution that gives the best policy of the agent for this game. **Note**: the value function visualization from the lab will no longer be feasible since the state space is no longer binary.

3. Answer to the following questions:

   (a) What is the observed best policy of the agent? How do you justify the resulting actions?

   (b) Consider a Q-Learning agent whose current state information would be given by the *observations* in the maze (i.e. 3 for SL and SR, 2 for any other state). Can the Q-Learning agent learn to solve this environment? Justify your answer.

Result presentation

- Submit the code used to implement the Point-based Value Iteration algorithm that helps you determine the agent policy.

- Submit a PDF file where you:
  - Provide a short README for the results of your PBVI implementation (Task 2)
  - Formalize the game as a POMDP (Task 1)
  - Provide the answers to the questions in Task 3