Assignment 2 - Theoretical Assignment
AI and Robotics (236609)

Prunotto Alice
alicep@campus.technion.ac.il
Computer Science Department
Technion - Israel Institute of Technology

Elashkin Andrew
swag@campus.technion.ac.il
Computer Science Department
Technion - Israel Institute of Technology

February 3, 2023

1 Theoretical Assignment

1.1 Modeling

1. For Classical Planning we will use Propositional STRIPS planning task defined by a tuple $P = \langle \mathcal{S}, \mathcal{I}, \mathcal{A}, \mathcal{G}, \mathcal{R} \rangle$, where

   - $\mathcal{S} \rightarrow$ states: there are 500 discrete states since there are 25 taxi positions, 5 possible locations of the passenger (including the case when the passenger is in the taxi), and 4 destination locations. Each state space is represented by the tuple: (taxi_row, taxi_col, passenger_location, destination).
   - $I \subseteq F \rightarrow$ initial state: random state from the state space.
   - $G \subseteq F \rightarrow$ goal state: when the position of the passenger = position of drop-off.
   - $\mathcal{A} \rightarrow$ set of actions: there are 6 discrete deterministic actions:
     $\mathcal{A} \in \{0 : move\_s, 1 : move\_n, 2 : move\_e, 3 : move\_w, 4 : pickup\_passenger, 5 : dropoff\_passenger\}$.
   - $\mathcal{R} \rightarrow$ reward: there are three different rewards: -1 per step unless other reward is triggered, +20 delivering passenger, -10 executing "pickup" and "drop-off" actions illegally.

2. Factored Markov-Decision Process (MDP) will be defined by the tuple $\langle \mathcal{S}, \mathcal{I}, \mathcal{A}, \mathcal{G}, \mathcal{R}, \mathcal{P}, \gamma \rangle$ where

   - $\mathcal{P} \rightarrow$ transition probability function: $\mathcal{P}(s, a, s')$, where s is the current state, a is the action and s' is the state you might reach with action a. It defines the probability of transitioning from one state to another given an action. There will be 1 probability for each possible movement (North, South, West, East) and 1 probability for pick-up and drop-off.
   - $\gamma \rightarrow$ discount factor: $\gamma \in [0, 1]$.

3. Partially Observable Markov Decision Process (POMDP) will be a tuple $\langle \mathcal{S}, \mathcal{I}, \mathcal{A}, \mathcal{G}, \mathcal{R}, \mathcal{P}, \gamma, \mathcal{O}, \Omega \rangle$ where

- $\Omega \rightarrow$ observation function: defines the probability of observing a particular observation given the current state and action, $\Omega = (North\_wall, South\_wall, East\_wall, West\_wall)$. When it observes a wall from some direction X, the wall is in direction X with probability 0.8 and with 0.2 probability the wall is to the right of that direction so, for example, if the taxi observes a wall to its north, it will obtain $\Omega = (0.8, 0.0, 0.2, 0.0)$.
- $\mathcal{O} \rightarrow$ set of observations: represents the information that the agent receives from the environment.

1.2 Best First Search

(a) $h_G(n)$ is not an admissible heuristic for the taxi domain. Considering the reward, the heuristic to be admissible needs to fulfill the following definition: $h'(n) \leq h(n)$.
In our problem, the heuristic $h_G(n)$ assigns a value of 1 to a node n that represents a terminal state, while the goal node has a reward value of +20. Since $+20 \leq 1$, the heuristic is not admissible. To be admissible it needs to assign to the terminal state a value that is greater than 20, so for example 30.

(b) Given the definition $h'(n) \leq h(n)$ for the heuristic $h'_G(n)$ to dominate $h_G(n)$, it is impossible to find an heuristic that both dominates and is admissible. Following this definition, the accumulated reward of the new heuristic should be less than the accumulated reward of the old one, but at the same time the old heuristic already had a value that was too little and did not over-estimates the total reward.
We can provide an admissible heuristic $h'_G(n)$ that does not dominate $h_G(n)$: we can assign +30 to the terminal state and +10 to every other state, in this way the accumulated reward is always over-estimated.

(c) No, the heuristic is not admissible for all deterministic and fully observable domains. It is only admissible for the specific taxi problem as it is defined. It assumes a specific reward structure and a specific layout of the grid-world environment, if something of these two are changed than it is not guaranteed that the heuristic will still be admissible.

(d) For an admissible heuristic for the stochastic version of the taxi domain we can use the same one we proposed in the answer (b) because in the stochastic version the expectation will always be lower than the value in the deterministic world, and so the reward will always be over-estimated.

1.3 Q Learning

1. GLIE principles can be supported in Q-learning by using a learning rate that decreases over time and an exploration policy that encourages more exploitation as the algorithm progresses. The first condition requires exploration to proceed indefinitely, while the second requires that its magnitude decays in time. We use the GLIE epsilon-greedy exploration principle, where on each time step t we select a random action with probability $p(t)$ and a greedy action with probability $1 - p(t)$. The probability $p(t) = 1/t$ for $t \rightarrow \infty$ will converge to zero. When t goes to infinity, the algorithm becomes completely greedy and exclusively exploits.

2. One alternative implementation of Q-learning that is GLIE is to use a Boltzmann exploration policy in combination with a gradually decreasing temperature parameter and by resetting the Q-values of all actions at the beginning of each episode.
   We select an action with probability $P(a|s) = \frac{exp(\frac{Q(s,a)}{T})}{\sum a' \in A exp(\frac{Q(s,a')}{T})}$, where T is the temperature. Large T means that each action has about the same probability, while small T leads to more greedy behavior. To satisfy the GLIE principles, the temperature parameter can be gradually decreased over time. This will encourage more exploitation of the learned Q-values as the algorithm progresses.
   In addition, the Q-learning algorithm can be modified to ensure that every action is selected at least once in every episode. This can be achieved by resetting the Q-values of all actions to a common initial value at the beginning of each episode. This will encourage the agent to try all actions at least once before making more informed decisions based on the learned Q-values.