National University of Singapore
School of Computing

Semester 2, AY2024-25

CS3263 Foundations of Artificial Intelligence

Issued: 20 March 2025 Due: 7 Aril 2025@23:59

Assignment 2

Programming Assignment: Your Mini House AI Assistant

[20 marks] Suppose you are designing an AI assistant for your house, and you wish to let the AI assistant help you rearrange the items in your house. Mini-house environment is a simple but scalable household environment in Python. A mini-house environment consists of rooms (e.g., bedrooms), moveable items (e.g., apples), containers (e.g., fridges), and surfaces (e.g., tables).

- **Rooms:** Each house comprises various distinct spaces like bedrooms, kitchens, dining rooms, living rooms, bathrooms, and hallways that interconnect these rooms.
- **Moveable Items:** The AI assistant has the capability to interact with and reposition items. Examples include fruits (e.g., apples, bananas), clothing (e.g., t-shirts), and other everyday items (e.g., shampoo).
- **Containers:** Representing smaller sections within rooms, containers (e.g., cabinets and fridges) can be opened or closed. The AI assistant can interact with items inside an open container but cannot alter the container's position.
- **Surfaces:** Representing open areas within rooms, surfaces (e.g., dinner tables or kitchen counters) allow the AI assistant to pick up items on the surfaces or place other items on the surfaces. However, their positions remain fixed.

Your AI assistant has pre-defined actions.

- **Pick:** Allows the AI assistant to grasp specific moveable items.
- **Place:** Grants the AI assistant the ability to set down items on a designated surface or inside a container.
- **Move:** Permits the AI assistant to transition between interconnected rooms. For instance, if the AI is in the bedroom, using the 'Move' action could allow it to navigate to an adjoining bathroom or hallway.

Due to the imperfect action execution, there will be uncertainty about the resulting state after your AI assistant applies actions to the environment. We can formulate the object-rearrangement problem as a Markov Decision Process. Assume the state $s \in \mathcal{S}$ is an object-centric state representation, denoting each object's position as well as the robot's position in the house. $a \in \mathcal{A}$ is a set of grounded actions. $\mathcal{R} : \mathcal{S} \to \mathbb{R}$ is the reward function. If the target object is in the desired state, you will receive a positive reward, or a small penalty otherwise. The transition $p(S' = s'|S = s, A = a)$ denotes the probability of the next state s' when applying action a at state s.

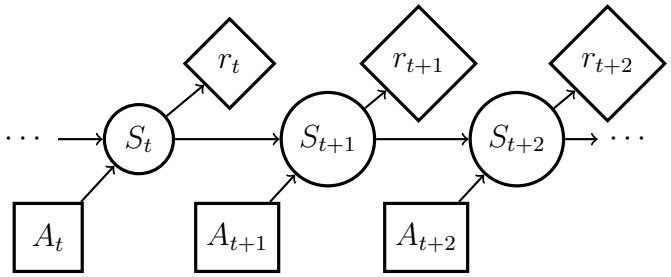

Figure 1: Markov Decision Process.

Question 1: AI Assistant as Markov Decision Process

Implement the **value iteration** (5 marks) and **policy iteration** (5 marks) algorithms for object rearrangements policy $\pi : \mathcal{S} \to \mathcal{A}$, such that it maximizes the accumulated reward. We provide the programming template for you to fill in your implementation. Please check it out at the official repository for details.

Question 2: AI Assistant as Reinforcement Learning

Implement the **Q-learning** (5 marks) algorithm for object rearrangements policy $\pi : \mathcal{S} \rightarrow \mathcal{A}$, such that it maximizes the accumulated reward. We provide the programming template for you to fill in your implementation. Please check it out at the official repository for details.

Question 3: AI Assistant in the Real World

Anwser the following questions (5 marks):

a) If the Minihouse has a **larger** environment consisting of 10 movable items and more rooms, can **value iteration** or **policy iteration** algorithms work? What other solution algorithm(s) would you use?

b) What about dealing with 100 moveable objects?

c) Can you use other model-free reinforcement learning algorithms to help in the cases above? Why or why not?

Use pre-trained **Generative Model APIs** to improve the capability of the Q-learning algorithm in solving large-scale decision-making problems. Please check it out at the official repository for details. Write your answer in the `Assignment2.ipynb` file in the GitHub repository.
