Planning, Learning and Decision Making

MSc in Computer Science and Engineering

First examination — April 27, 2023

Instructions

- You have 120 minutes to complete the examination.
- Make sure that your test has a total of 10 pages and is not missing any sheets, then write your full name and student n. on this page (and your number in all others).
- The test has a total of 5 questions, with a maximum score of 20 points. The questions have different levels of difficulty. The point value of each question is provided next to the question number.
- *If you get stuck in a question, move on.* You should start with the easier questions to secure those points, before moving on to the harder questions.
- *No interaction with the faculty is allowed during the exam.* If you are unclear about a question, clearly indicate it and answer to the best of your ability.
- Please provide your answer in the space below each question. If you make a mess, clearly indicate your answer.
- The exam is open book and open notes. You may use a calculator, but any other type of electronic or communication equipment is not allowed.
- Good luck.

Question 1. (2.5 pts.)

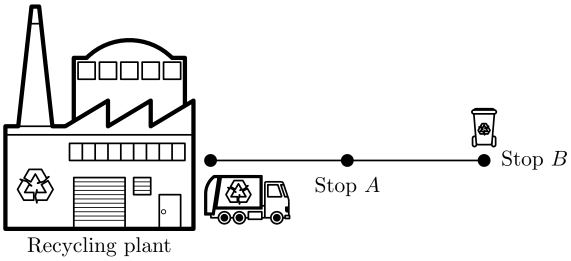


Figure 1: The garbage truck must collect garbage from stops A and B.

Consider the diagram in Fig. 1, representing a scenario similar to those you encountered in the lab assignments. The building on the left corresponds to a recycling plant. A truck leaves the recycling plant and must traverse the road network depicted in the diagram to collect garbage at stop B. After collecting the garbage, the truck must return to the recycling plant to drop it.

In each location the driver has four actions available (all actions take 10 minutes to complete):

- *Collect garbage.* There may or may not be garbage to collect in stop B. When the truck is stop B, there is garbage to collect, and the truck is empty, the action successfully collects the garbage. Otherwise, the action has no effect. A successful garbage collection implies that: (i) there will be no garbage to collect in the next time step; and (ii) the truck becomes full.
- *Drop garbage.* In stops A and B, this action has no effect. In the recycling plant, this action successfully drops collected garbage *if the truck is full.* Otherwise, it has no effect.
- *Move left.* In the recycling plant, this action has no effect. In all other locations, it moves the truck to the adjacent location to the left.
- *Move right.* In stop B, this action has no effect. In all other locations, it moves the truck to the adjacent location to the right.

At each time step t, there is a 0.3 probability that garbage will appear in stop B if there was no garbage there at time step $t - 1$. If garbage appears in stop B, it will remain there until it is collected by the truck. However, the driver cannot observe whether there is garbage in stop B unless if it goes there.

For every 10 minutes that the garbage remains uncollected there is a cost of 0.5 (other periods of time correspond to proportional costs), unless if a successful garbage drop takes place, in which case no cost is incurred. In any state, performing an action with no effect has maximum cost.

Consider that a new time step occurs whenever the driver takes an action at any of the three locations (Recycling plant and stops A and B).

Describe the decision problem faced by the agent using the adequate type of model, indicating:

- The type of model needed to describe the decision problem of the agent;
- The state, action, and observation spaces (when relevant);
- The transition probabilities corresponding to the action "Move Left";
- The immediate cost function.

Solution 1.

To choose what actions to take, the agent (i.e., the truck driver) should know its position (Recycling plant, stop A or stop B), whether the truck is loaded or not, and whether there is garbage in stop B or not. The state should, then, contain this information. However, the driver cannot observe whether there is garbage in stop B or not unless if it goes there. As such, the problem has *partial observability*.

We thus model the problem as a POMDP $(\mathcal{X}, \mathcal{A}, \mathcal{Z}, \{\mathbf{P}_a\}, \{\mathbf{O}_a\}, c, \gamma)$, where

- The state space consists of all triplets (p, l, g), with $p \in \{R, A, B\}$ and representing the position of the truck; $l \in \{0, 1\}$ denoting whether the truck is loaded or not; and $g \in \{0, 1\}$ denoting whether there is garbage in stop B or not. We thus have:

$$\begin{aligned}\mathcal{X} = \{&(R, 0, 0), (R, 0, 1), (R, 1, 0), (R, 1, 1),\\ &(A, 0, 0), (A, 0, 1), (A, 1, 0), (A, 1, 1),\\ &(B, 0, 0), (B, 0, 1), (B, 1, 0), (B, 1, 1)\}\end{aligned}$$

- The action space is $\mathcal{A} = \{C, D, L, R\}$, corresponding to the actions *Collect garbage*, *Drop garbage*, *Move left*, and *Move right*, respectively.

- The observation space consists of triplets (p, l, g) such that $p \in \{R, A, B\}$ and corresponds to the position of the truck; $l \in \{0, 1\}$ and denotes whether the truck is loaded or not; and $g \in \{\emptyset, 0, 1\}$ and indicates whether the driver observes garbage in the current location (1), no garbage in the current location (0), or nothing ($\emptyset$). We thus have:

$$\begin{aligned}\mathcal{Z} = \{&(R, 0, \emptyset), (R, 1, \emptyset), (A, 0, \emptyset), (A, 1, \emptyset),\\ &(B, 0, 0), (B, 0, 1), (B, 1, 0), (B, 1, 1)\}\end{aligned}$$

- The transition probabilities for the action L can be summarized in the matrix:

$$\mathbf{P}_L = \begin{bmatrix}
0.7 & 0.3 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0\\
0.0 & 1.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0\\
0.0 & 0.0 & 0.7 & 0.3 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0\\
0.0 & 0.0 & 0.0 & 1.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0\\
0.7 & 0.3 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0\\
0.0 & 1.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0\\
0.0 & 0.0 & 0.7 & 0.3 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0\\
0.0 & 0.0 & 0.0 & 1.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0\\
0.0 & 0.0 & 0.0 & 0.0 & 0.7 & 0.3 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0\\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0\\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.7 & 0.3 & 0.0 & 0.0 & 0.0 & 0.0\\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.0 & 0.0 & 0.0 & 0.0 & 0.0
\end{bmatrix}.$$

- Finally, the immediate cost function c can be represented as a matrix

$$c = \begin{bmatrix} 1.0 & 1.0 & 1.0 & 0.0 \\ 1.0 & 1.0 & 1.0 & 0.5 \\ 1.0 & 0.0 & 1.0 & 0.0 \\ 1.0 & 0.0 & 1.0 & 0.5 \\ 1.0 & 1.0 & 0.0 & 0.0 \\ 1.0 & 1.0 & 0.5 & 0.5 \\ 1.0 & 1.0 & 0.0 & 0.0 \\ 1.0 & 1.0 & 0.5 & 0.5 \\ 1.0 & 1.0 & 0.0 & 1.0 \\ 0.5 & 1.0 & 0.5 & 1.0 \\ 1.0 & 1.0 & 0.0 & 1.0 \\ 1.0 & 1.0 & 0.5 & 1.0 \end{bmatrix}.$$

In the remainder of the test, consider the POMDP $\mathcal{M} = (\mathcal{X}, \mathcal{A}, \mathcal{Z}, \{\mathbf{P}_a\}, \{\mathbf{O}_a\}, c, \gamma)$ where

- $\mathcal{X} = \{A, B, C\}$;
- $\mathcal{A} = \{a, b, c\}$;
- $\mathcal{Z} = \{u, v\}$;
- The transition probabilities are

$$\mathbf{P}_a = \begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 0.2 & 0.8 \\ 0.0 & 0.8 & 0.2 \end{bmatrix}; \qquad \mathbf{P}_b = \begin{bmatrix} 0.2 & 0.8 & 0.0 \\ 0.8 & 0.2 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}; \qquad \mathbf{P}_c = \begin{bmatrix} 0.2 & 0.0 & 0.8 \\ 0.0 & 1.0 & 0.0 \\ 0.8 & 0.0 & 0.2 \end{bmatrix}.$$

- The observation probabilities are

$$\mathbf{O}_a = \mathbf{O}_b = \mathbf{O}_c = \begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \\ 1.0 & 0.0 \end{bmatrix}.$$

- The cost function c is given by

$$\mathbf{C} = \begin{bmatrix} 0.5 & 0.0 & 1.0 \\ 0.0 & 1.0 & 0.5 \\ 1.0 & 0.5 & 0.0 \end{bmatrix}.$$

- Finally, the discount is given by $\gamma = 0.9$.

Question 2. (**8 pts.**)

For each of the following questions, indicate the *single most correct answer*.

(a) **(0.8 pts.)** Consider the MDP obtained from $\mathcal{M}$ by ignoring partial observability, which corresponds to the tuple $(\mathcal{X}, \mathcal{A}, \{\mathbf{P}_a\}, c, \gamma)$. Which stationary policy π induces an irreducible Markov chain $(\mathcal{X}, \mathbf{P}_\pi)$?

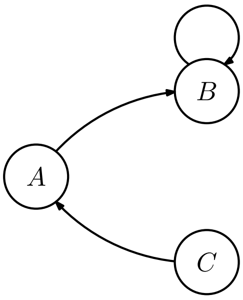
□ $\pi = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}.$

□ $\pi = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$

□ $\pi = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}.$

☒ **None of the above.**

(b) **(0.8 pts.)** Consider the Markov chain



The Markov chain has...

□ One communicating class.

□ Two communicating classes.

☒ **Three communicating classes.**

□ It is not possible to determine the number of communicating classes from the diagram alone.

(c) **(0.8 pts.)** Consider a set of outcomes $\mathcal{X} = \{x, y, z\}$ and a preference relation $\succ$ on such that $x \succ y \succ z$. A utility function to represent the previous relation could be...

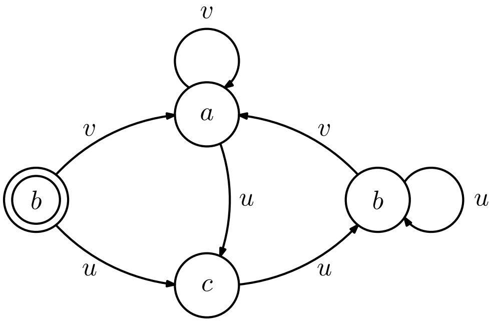
□ $u(x) = 1$, $u(y) = 2$, $u(z) = 3$.

☒ $u(x) = 3$, $u(y) = 2$, $u(z) = 1$.

□ The relation $\succ$ is not rational, so it cannot be translated to a utility function.

□ None of the above.

(d) **(0.8 pts.)** Consider the POMDP $\mathcal{M}$ and suppose that the corresponding optimal policy can be represented as the policy graph:



The double-circled node corresponds to the empty history. Suppose that the history at time step $t = 3$ is $h_3 = \{b, v, a, v, a, u\}$. The optimal action is, then,

□ $a_3 = a$.

□ $a_3 = b$.

☒ $a_3 = c$.

□ There is not enough information to compute the optimal action at time step $t = 3$.

(e) **(0.8 pts.)** Inverse reinforcement learning is an ill-posed problem because...

□ ... we need to learn a policy from examples by a teacher.

□ ... it does not consider the possibility that the teacher can make mistakes.

□ ... it does not use a Bayesian approach.

☒ **None of the above.**

(f) **(0.8 pts.)** In inverse reinforcement learning, if the cost function to be computed can be arbitrary, ...

□ ... IRL algorithms learn faster.

□ ... IRL algorithms learn slower.

☒ **... the teacher must demonstrate the optimal action in all states for the cost function to be recovered.**

□ None of the above.

(g) **(0.8 pts.)** The UCB algorithm makes use of...

☒ **... the principle of optimism in the face of uncertainty (unknown is probably better).**

□ ... the principle of Occam's razor (simpler models are preferable to complex models).

□ ... the inductive learning assumption (if we learn from enough examples, we'll perform well in the actual task).

□ None of the above.

(h) **(0.8 pts.)** In an adversarial multi-armed bandit problem...

□ ... the best algorithm to use is EWA.

□ ... the best algorithm to use is the weighted majority algorithm.

□ ... the best algorithms to use are point-based methods like PERSEUS.

☒ **None of the above.**

(i) **(0.8 pts.)** Consider the update for REINFORCE:

$$w_{t+1} = w_t - \alpha_t \left[\sum_{t=0}^{N} \nabla_w \log \pi_{w_t}(x_{t+n}, a_{t+n}) \sum_{m=n}^{N} \gamma^m c_{t+n} \right],$$

where w_t is the parameter of the policy, and the summation is computed from a trajectory $\{x_t, a_t, c_t, x_{t+1}, \ldots, c_{t+N}, x_{t+N+1}\}$ obtained with policy π_{w_t}.

☒ **The REINFORCE algorithm can be considered an *on-policy algorithm.***

□ The REINFORCE algorithm can be considered an *off-policy algorithm.*

□ REINFORCE can be considered *neither* on-policy nor off-policy, since it is a policy gradient algorithm.

□ REINFORCE can be considered *both* on-policy and off-policy, since it is a policy gradient algorithm.

(j) **(0.8 pts.)** The use of a baseline in the REINFORCE algorithm...

□ ... helps to mitigate the bias in the gradient estimate.

□ ... is used to improve the data efficiency the algorithm.

☒ **... is used to decrease the variance of the gradient estimate.**

□ All of the above.

Question 3. (**4 pts.**)
Consider the MDP $(\mathcal{X}, \mathcal{A}, \{\mathbf{P}_a\}, c, \gamma)$, obtained from $\mathcal{M}$ by ignoring partial observability. Consider also the policy

$$\pi = \begin{bmatrix} 0.0 & 0.5 & 0.5 \\ 1.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 \end{bmatrix}.$$

(a) **(2.5 pt.)** Compute J^π.

(b) **(1.5 pts.)** Perform one step of policy improvement given J^π from (a).

Note: If you did not solve (a), you can use

$$\mathbf{J}^\pi = \begin{bmatrix} 4.5 & 4.0 & 5.0 \end{bmatrix}^\top.$$

Solution 3.

(a) To compute J^π, we start by computing $\mathbf{P}_\pi$ and c_π. We have that

$$\mathbf{P}_\pi = \begin{bmatrix} 0.2 & 0.4 & 0.4 \\ 0.0 & 0.2 & 0.8 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}, \qquad \mathbf{c}_\pi = \begin{bmatrix} 0.5 \\ 0.0 \\ 0.5 \end{bmatrix}.$$

It follows that

$$\begin{aligned} \mathbf{J}^\pi &= (\mathbf{I} - \gamma \mathbf{P}_\pi)^{-1} \mathbf{c}_\pi \\ &= \left(\begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix} - 0.9 \begin{bmatrix} 0.2 & 0.4 & 0.4 \\ 0.0 & 0.2 & 0.8 \\ 0.0 & 0.0 & 1.0 \end{bmatrix} \right)^{-1} \begin{bmatrix} 0.5 \\ 0.0 \\ 0.5 \end{bmatrix} \\ &= \begin{bmatrix} 0.82 & -0.36 & -0.36 \\ 0.0 & 0.82 & -0.72 \\ 0.0 & 0.0 & 0.1 \end{bmatrix}^{-1} \begin{bmatrix} 0.5 \\ 0.0 \\ 0.5 \end{bmatrix} \\ &= \begin{bmatrix} 4.73 \\ 4.39 \\ 5.0 \end{bmatrix}. \end{aligned}$$

(b) To perform one step of policy optimization, we first compute Q^π from J^π, where

$$\mathbf{Q}^\pi(:, a) = \mathbf{c}_a + \gamma \mathbf{P}_a \mathbf{J}^\pi, \qquad \text{for all } a \in \mathcal{A}.$$

We thus have:

$$\mathbf{Q}^\pi(:, a) = \mathbf{c}_a + \gamma \mathbf{P}_a \mathbf{J}^\pi = \begin{bmatrix} 4.76 \\ 4.39 \\ 5.06 \end{bmatrix},$$

$$\mathbf{Q}^\pi(:, b) = \mathbf{c}_b + \gamma \mathbf{P}_b \mathbf{J}^\pi = \begin{bmatrix} 4.01 \\ 5.20 \\ 5.0 \end{bmatrix},$$

$$\mathbf{Q}^\pi(:, c) = \mathbf{c}_c + \gamma \mathbf{P}_c \mathbf{J}^\pi = \begin{bmatrix} 5.45 \\ 4.45 \\ 4.3 \end{bmatrix},$$

yielding

$$\mathbf{Q}^\pi = \begin{bmatrix} 4.76 & 4.01 & 5.45 \\ 4.39 & 5.20 & 4.45 \\ 5.06 & 5.0 & 4.3 \end{bmatrix}.$$

The greedy policy with respect to J^π thus comes

$$\pi_g^{J^\pi} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Question 4. (3.5 pts.)

Consider once again the MDP $(\mathcal{X}, \mathcal{A}, \{\mathbf{P}_a\}, c, \gamma)$, obtained from $\mathcal{M}$ by ignoring partial observability, and let π denote the policy π from Question 3:

$$\pi = \begin{bmatrix} 0.0 & 0.5 & 0.5 \\ 1.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 \end{bmatrix}.$$

Suppose that the agent wants to learn Q^π using reinforcement learning, for which it uses a novel algorithm that, at each time step t, takes a transition (x_t, a_t, c_t, x_{t+1}) and updates the estimate for Q^π as

$$Q_{t+1}(x_t, a_t) = Q_t(x_t, a_t) + \alpha_t(c_t + \gamma\mathbb{E}_{\mathrm{a}\sim\pi(x_{t+1})}[Q_t(x_{t+1}, \mathrm{a})] - Q_t(x_t, a_t)),$$

where π in the expectation is the policy to be evaluated. Further assume that the agent interacts with the environment and experiences the following transitions:

$$(A, a, 0.5, A), (A, c, 1.0, A), ...$$

(a) **(2.0 pts.)** Assuming that $Q_0(x, a) \equiv 0$, use the transitions above to perform two updates using the algorithm above. Use a step-size $\alpha = 0.2$.

(b) **(0.75 pt.)** The algorithm above is called *Expected-SARSA*. Is Expected-SARSA an on-policy algorithm or an *off-policy* algorithm? Explain your reasoning.

(c) **(0.75 pt.)** How could Expected-SARSA be used to compute Q^*?

Solution 4.

(a) The first update comes:

$$\begin{aligned} Q_1(A, a) &= Q_0(A, a) + \alpha(0.5 + 0.9\,\mathbb{E}_{\mathrm{a}\sim\pi(A)}[Q_0(A, \mathrm{a})] - Q_0(A, a)) \\ &= 0.0 + 0.2(0.5 + 0.9 \cdot 0.0 - 0.0) = 0.1 \end{aligned}$$

$$\begin{aligned} Q_2(A, c) &= Q_1(A, c) + \alpha(1.0 + 0.9\,\mathbb{E}_{\mathrm{a}\sim\pi(A)}[Q_1(A, \mathrm{a})] - Q_1(A, c)) \\ &= 0.0 + 0.2(1.0 + 0.9 \cdot 0.0 - 0.0) = 0.2. \end{aligned}$$

(b) Expected-SARSA is an off-policy algorithm, since it computes the Q-values for policy π independently of the behavior policy used (i.e., of the policy used to interact with the environment). For example, in the computations above, we did not require any knowledge of the behavior policy, and used only the policy π to be evaluated.

(c) Expected-SARSA could be used to compute Q^* much in the same way as SARSA can—by using it as the evaluation step in (generalized) policy iteration. In particular, once Q^π is computed, an improved policy π' is computed from Q^π. This new policy is evaluated (using, once again, Expected-SARSA) and the process repeats.

Question 5. (**2 pts.**)

In HMMs, the *forward mapping* at time step t is defined as a mapping $\alpha_t : \mathcal{X} \to \mathbb{R}$ such that

$$\alpha_t(x) = \mathbb{P}_{\mu_0}\left[\mathrm{x}_t = x, \mathbf{z}_{0:t} = \boldsymbol{z}_{0:t}\right],$$

where μ_0 is the HMM's initial distribution. Show that α_t verifies the recursive relation

$$\alpha_{t+1}(x') = \sum_{x \in \mathcal{X}} \alpha_t(x) \mathbf{P}(x' \mid x) \mathbf{O}(z_{t+1} \mid x').$$

Solution 5.

By definition,

$$\begin{aligned}\alpha_{t+1}(x') &= \mathbb{P}_{\mu_0}\left[\mathrm{x}_{t+1} = x', \mathbf{z}_{0:t+1} = \boldsymbol{z}_{0:t+1}\right] \\ &= \mathbb{P}_{\mu_0}\left[\mathrm{x}_{t+1} = x', \mathbf{z}_{0:t} = \boldsymbol{z}_{0:t}, \mathrm{z}_{t+1} = z_{t+1}\right].\end{aligned}$$

From the definition of conditional probability,

$$\alpha_{t+1}(x') = \mathbb{P}_{\mu_0}\left[\mathrm{z}_{t+1} = z_{t+1} \mid \mathrm{x}_{t+1} = x', \mathbf{z}_{0:t} = \boldsymbol{z}_{0:t}\right] \mathbb{P}_{\mu_0}\left[\mathrm{x}_{t+1} = x', \mathbf{z}_{0:t} = \boldsymbol{z}_{0:t}\right].$$

Since the observations are independent of the history given the state, the above expression simplifies to

$$\alpha_{t+1}(x') = \mathbf{O}(z_{t+1} \mid x') \mathbb{P}_{\mu_0}\left[\mathrm{x}_{t+1} = x', \mathbf{z}_{0:t} = \boldsymbol{z}_{0:t}\right].$$

Equivalently,

$$\alpha_{t+1}(x') = \mathbf{O}(z_{t+1} \mid x') \sum_{x \in \mathcal{X}} \mathbb{P}_{\mu_0}\left[\mathrm{x}_{t+1} = x', \mathrm{x}_t = x, \mathbf{z}_{0:t} = \boldsymbol{z}_{0:t}\right].$$

Again using the definition of conditional probability, we get

$$\begin{aligned}\alpha_{t+1}(x') &= \mathbf{O}(z_{t+1} \mid x') \sum_{x \in \mathcal{X}} \mathbb{P}_{\mu_0}\left[\mathrm{x}_{t+1} = x' \mid \mathrm{x}_t = x, \mathbf{z}_{0:t} = \boldsymbol{z}_{0:t}\right] \mathbb{P}_{\mu_0}\left[\mathrm{x}_t = x, \mathbf{z}_{0:t} = \boldsymbol{z}_{0:t}\right] \\ &= \mathbf{O}(z_{t+1} \mid x') \sum_{x \in \mathcal{X}} \mathbf{P}(x' \mid x) \alpha_t(x),\end{aligned}$$

where we used the Markov property and the definition of $\alpha_t(x)$.