Reinforcement Learning

Introduction

Lecturer: *Nicolò Cesa-Bianchi* version January 20, 2023

This material is partially based on the book draft "Reinforcement Learning: Foundations" by Shie Mannor, Yishay Mansour, and Aviv Tamar.

Reinforcement Learning (RL) is concerned with the design and analysis of algorithms that learn how to make decisions in arbitrary environments. A crucial aspect is that decisions must be taken sequentially and the algorithms must consider the implications of their decisions. Some practical problems to which RL can be applied are:

- Playing a board game, like Chess or Go.
- Controlling a robot to achieve a certain task; for example, collecting mineral samples or rescuing people.
- Driving a car to a given destination.
- Keeping the parameters of a physical process in a safe and useful range of values (e.g., a controlled nuclear reaction that generates heat).
- Deciding which advertisement to show to each new visitor of a website.
- Managing an investment fund.

There are two features that typically distinguish RL applications from standard ML applications:

- The decisions made by the algorithm may affect the outcome of future decisions.
- When making a decision among a number of options, the algorithm typically observes only the outcome of the chosen option; the outcomes of the other options remain partially or totally unknown.

Note that only one of these two features may appear in a RL application. For example, in advertising we may ignore the effect on the next visitor of showing an ad to the current visitor. On the other hand, we can simulate the outcome of an investment decision irrespective to the decision we actually made.

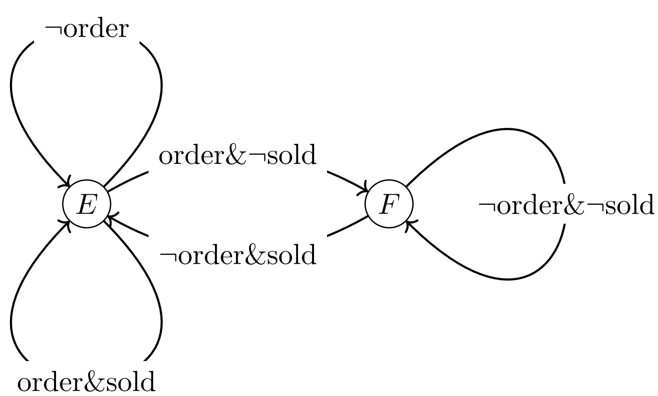
We use the word agent to refer to the algorithm operating in an environment. We can abstract the interaction between agent and environment through the mathematical notion of **discrete-time decision process**. At each time step, the agent is in a known state and the following happens:

1. The agent selects an **action** among those available in the current state and executes it
2. The agent's current **state** changes.

In this course we focus on **finite decision processes**, where the set $\mathcal{S}$ of states and the set $\mathcal{A}$ of actions are both finite. For each $s \in \mathcal{S}$, we use $\mathcal{A}(s)$ to denote the actions available in state s.

Example 1 (Finding the shortest path in a weighted directed graph.) *This is one of the most fundamental algorithmic problems on graphs and can be viewed as a deterministic decision process where the states are the nodes of the graph and the actions available in each state correspond to the outgoing edges. The edge weight is the cost of traversing the edge. Given a start state and a goal state, the agent must find the sequence of actions corresponding to the path of minimum total cost from the start node to the goal node. When the graph is fully known, the optimal sequence of actions can be found using, for example, Dijkstra's algorithm for single source and single destination shortest path.*

Example 2 (Managing an inventory.) *A retailer sells one item at the time of a certain set of goods. The retailer either has not items in stock (state E) or has exactly one item is stock (state F). If the state is E, the retailer can order one item from the supplier (action "order"). If the item is immediately sold, then the next state is again E; otherwise, the next state is F. If the state is F and the item in stock gets sold, then the next state is E. If the item remains in stock, then the retailer pays for holding the stock. Note that action "¬order" (do not order a new item) is the only action possible in state F. If action "¬order" is executed in state E, then the retailer may miss a sale if someone willing to buy shows up.*



In order to capture the uncertainty in the effect of an action on the environment, we allow the state transition to be stochastic, where the distribution over the future state depends on both the current state and the action selected by the agent. In symbols, for every $s \in \mathcal{S}$ and $a \in \mathcal{A}(s)$ there exists a probability distribution $p(\cdot \mid s, a)$ over $\mathcal{S}$ where $p(s' \mid s, a)$ denotes the **transition kernel**, that is the probability that the next state is s' when action a is executed in state s.

The triple $\big\langle \mathcal{S}, \mathcal{A}, \{p(\cdot \mid s,a) \,:\, s \in \mathcal{S},\, a \in \mathcal{A}(s)\} \big\rangle$ defines a **Markov Decision Process** (MDP). The Markovian property refers to the fact that the next state only depend on the current state and the selected action, and not to the previous states and actions.

The behavior of the agent interacting with an MDP is specified by a **control policy**. A deterministic control policy maps states to actions. The stochastic sequence of states and actions generated by a policy started from some initial state $s_0 \in \mathcal{S}$ is called a trajectory. A policy $\boldsymbol{\pi} = (\pi_t)_{t\geq 0}$ is a sequence of mappings π_t, where each π_t maps any possible history (i.e., past trajectory) $\boldsymbol{h}_t = (s_0, a_0, \dots, s_{t-1}, a_{t-1}, s_t)$ to an action $a_t \in \mathcal{A}(s_t)$. A deterministic **Markov policy** $\boldsymbol{\pi} = (\pi_t)_{t\geq 0}$ can be written, for all $t \geq 1$, as $\pi_t : \mathcal{S} \to \mathcal{A}$ such that $a_t = \pi_t(s_t)$. In other words, the action selected at time t only depends on the current state s_t and not on the history.

Control policies can be **randomized**. Then $\pi_t(\boldsymbol{h}_t)$ is a probability distribution over $\mathcal{A}(s_t)$ and we write $\pi_t(a \mid \boldsymbol{h}_t)$ to denote $\mathbb{P}\big(a_t = a \mid \boldsymbol{h}_t\big)$, where we are conditioning on the history $\boldsymbol{h}_t$.

Recall that a **Markov chain** on a state space $\mathcal{S}$ with initial state s_0 is a random walk $s_0, s_1, \dots$ over $\mathcal{S}$ such that

$$\mathbb{P}\big(s_t = s' \mid s_0, \dots, s_{t-1}\big) = \mathbb{P}\big(s_t = s' \mid s_{t-1}\big)$$

for all $s' \in \mathcal{S}$ and for all $t \geq 1$.

If we fix a Markov policy and an initial state $s_0 \in \mathcal{S}$, then the stochastic sequence $(s_t)_{t\geq 0}$ of states traversed by the policy is a Markov chain with transition probabilities

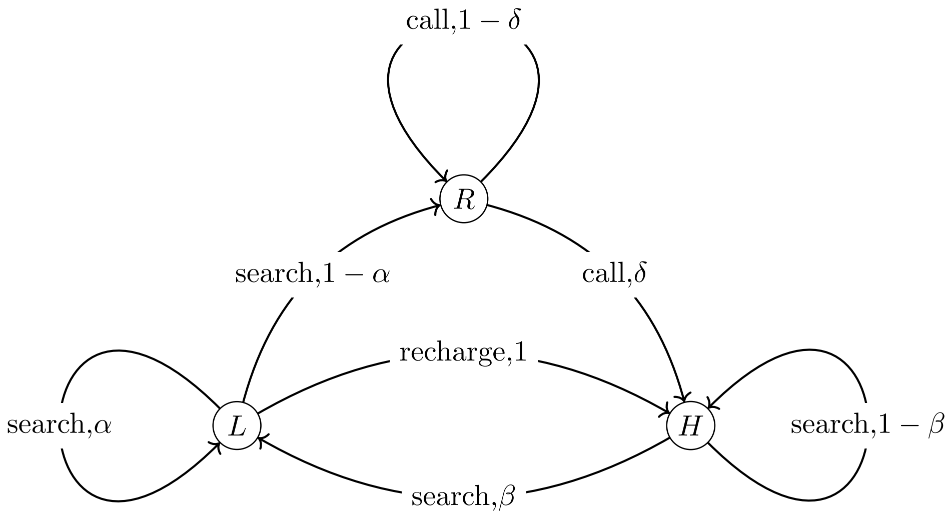
$$\begin{aligned}
\mathbb{P}\big(s_{t+1} = s' \mid s_t = s\big) &= \sum_{a\in\mathcal{A}_t} \mathbb{P}\big(s_{t+1} = s',\, a_t = a \mid s_t = s\big) \\
&= \sum_{a\in\mathcal{A}_t} \mathbb{P}\big(s_{t+1} = s' \mid s_t = s,\, a_t = a\big)\mathbb{P}(a_t = a \mid s_t = s) \\
&= \sum_{a\in\mathcal{A}_t} p(s' \mid s, a)\pi_t(a \mid s)
\end{aligned}$$

where here and in what follows we write $\mathcal{A}_t$ instead of $\mathcal{A}(s_t)$.

Recall that an MDP models an environment, while the policy defines a behavior of the agent. In general, we would like the agent to behave in certain way. This corresponds to finding the "right" policy for the environment. Thus, we must find a way of assigning values to the policies, so that the agent can learn the policy with the highest value, which is the best one for the given environment.

As policies define stochastic trajectories $(s_0, a_0, s_1, a_1, \dots)$ of state and actions, we want to assign a value to each trajectory. In the MDP framework, this is done by assigning a reward $r_t(s, a)$ to each state-action pair where $r_t : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is a time-dependent **reward function**. In many practical examples, the interaction between the agent and the environment ends after a certain number of rounds (e.g., in a Chess game). In this case, we also use a **terminal reward function** $r_T : \mathcal{S} \to \mathbb{R}$ assigning a value to the state reached at the end of the interaction (in a Chess game, the terminal reward reflects whether the agent won the game whereas the rewards assigned to state-action pairs may help the agent distinguish "good" moves from "bad" moves).

Example 3 (Recycling robot.) *A robot roams an office to collect empty cans that have to be recycled. The robot can be in three states depending on the battery charge: H (high charge), L (low charge), R (rescue me). In states H and L, the robot can search for cans and get a positive reward $r_s > 0$. In state L, instead of searching the robot can recharge and obtain a zero reward. In state R the robot can only call for help, each time getting a negative reward $r_c < 0$. In the figure below, the label a, p on an edge (s, s') indicates the action name a and the transition probability $p = p(s' \mid s, a)$.*



The role of the reward is similar to that of the loss in supervised machine learning: the reward is the only way through which the agent can learn a desired behavior. In particular, we design agents that seek to maximize their return (i.e., expected cumulative reward). Given a policy with stochastic trajectory $(s_0, a_0, s_1, a_1, \ldots)$ on a given MDP, we can define the return according to the two **evaluation criteria**:

- Finite horizon: $\mathbb{E}\left[\sum_{t=0}^{T-1} r_t(s_t, a_t) + r_T(s_T)\right]$
- Infinite horizon: $\lim_{T\to\infty} \mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T-1} r_t(s_t, a_t)\right]$

In the case of infinite horizon, clearly there is no terminal reward.

We also consider the important case in which T can be a random variable. To define the return using a stochastic horizon, we identify a subset $\mathcal{G} \subseteq \mathcal{S}$ of **goal states** in the MDP and stop the interaction as soon as the agent reaches any state in $\mathcal{G}$. Denoting by $s_0, s_1, \ldots$ the stochastic trajectory of states realized by the agent's policy, we define $T = \min\{t \geq 0 \,:\, s_t \in \mathcal{G}\}$. For technical reasons, we assume that

$$\sum_{t=0}^{\infty} \mathbb{P}(T \geq t) < \infty$$

It is easy to see that any MDP with state space $\mathcal{S}$ and finite horizon can be transformed into an equivalent MDP with state space $\mathcal{S} \times [T]$, goal states $\mathcal{G} = \{(s, T) \,:\, s \in \mathcal{S}\}$ and stochastic horizon $T = \min\{t \geq 0 \,:\, s_t \in \mathcal{G}\}$. Hence the finite horizon is a special case of the stochastic horizon.

If the MDP has no goal states, then a policy may run forever and the only meaningful evaluation criterion is the infinite criterion. However, we can always make a small modification to the MDP so that the horizon becomes stochastic rather than infinite. Given an MDP $\big\langle \mathcal{S}, \mathcal{A}, p(\cdot \mid s,a)\big\rangle$ without goal states, we add a single goal state s_G and define a new transition kernel p' defined by $p'(s_G \mid s,a) = 1-\gamma$ and $p'(s' \mid s,a) = \gamma\, p(s' \mid s,a)$ for each $s, s' \in \mathcal{S} \setminus \{s_G\}$, $a \in \mathcal{A}$, and for some $0 < \gamma < 1$. The resulting evaluation criterion, which is a special case of stochastic horizon, is known as the **discounted infinite horizon** (or γ-discounted horizon). Intuitively, the discounted infinite horizon assumes that the interaction may stop at any point of time with probability γ. In practice, a potentially infinite interaction may stop because, for example, the hardware (e.g., a robot or a car) executing the policy breaks down or is being serviced.

For all $t \geq 1$, assuming $s_{t-1} \neq s_G$, the probability of not stopping at time t is

$$\mathbb{P}\big(s_t \neq s_G \mid s_{t-1}, a_{t-1}\big) = \gamma \sum_{s' \neq s_G} p\big(s' \mid s_{t-1}, a_{t-1}\big) = \gamma$$

for any action $a_{t-1} \in \mathcal{A}_{t-1}$. Therefore, given any policy $\boldsymbol{\pi}$ executing actions $a_t = \pi_t(\boldsymbol{h}_t)$, the probability of not stopping before or at time t assuming $s_0 \neq s_G$ is

$$\mathbb{P}(T \geq t) = \mathbb{P}\big(s_1 \neq s_G, \ldots, s_t \neq s_G \mid s_0, a_0\big) = \prod_{t=1}^{T} \mathbb{P}\big(s_t \neq s_G \mid s_{t-1}, a_{t-1}\big) = \gamma^t$$

Since $T = t$ does not depend on the specific trajectory of states and actions, we can fix any infinite trajectory $\boldsymbol{h} = \big(s_0, a_0, s_1, a_1, \ldots\big)$ and write the expected return on this trajectory as

$$\begin{aligned} r_0(s_0,a_0) + \sum_{t=1}^{\infty}\sum_{\tau=1}^{t} r_\tau(s_\tau,a_\tau)\mathbb{P}(T=t) &= r_0(s_0,a_0) + \sum_{\tau=1}^{\infty} r_\tau(s_\tau,a_\tau) \sum_{t=\tau}^{\infty} \mathbb{P}(T=t) \\ &= r_0(s_0,a_0) + \sum_{\tau=1}^{\infty} r_t(s_\tau,a_\tau)\mathbb{P}(T \geq t) \\ &= \sum_{t=0}^{\infty} \gamma^t r_t(s_t,a_t) \end{aligned}$$

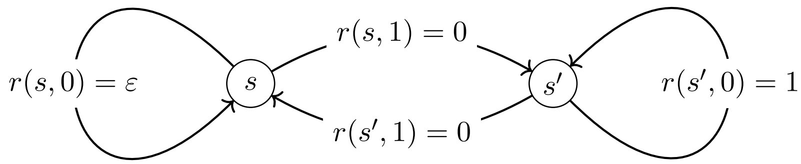
By now taking expectation with respect to the trajectory, we obtain that the return with respect to the discounted infinite horizon is

$$\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t(s_t,a_t)\right]$$

Suboptimality of the greedy policy. The greedy policy is a deterministic policy defined as

$$\pi_t(s) = \operatorname*{argmax}_{a \in \mathcal{A}} r_t(s,a)$$

It is easy to construct examples of MDP where this policy is never optimal. In the MDP below, if $s_0 = s$, the greedy policy achieves an expected return of ε with respect to the infinite horizon criterion. On the other hand, the optimal policy π^*, such that $\pi^*(s) = 1$ and $\pi^*(s') = 0$, achieves an expected return of 1 with respect to the same criterion.



We now show that in order to maximize any performance criterion it is sufficient to consider Markov policies. Let μ be a probability distribution over the initial state s_0 and let $q_t^{\boldsymbol{\pi}}(s,a) = \mathbb{P}^{\boldsymbol{\pi}}(s_t = a, a_t = a,\, T \geq t)$ be the **occupancy measure** evaluated at (s,a). This is the distribution of (s_t, a_t) under strategy $\boldsymbol{\pi}$ (with initial state distribution μ). We now show that the stochastic horizon performance criterion depends linearly on the rewards $r_t(s_t, a_t)$, which implies that any two policies that induce the same occupancy measure for all $t \geq 0$ have the same performance. Indeed,

$$\begin{aligned}\mathbb{E}\left[\sum_{t=0}^{T} r(s_t, a_t)\right] &= \sum_{t=0}^{\infty}\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} r_t(s,a)\mathbb{P}^{\boldsymbol{\pi}}(s_t = a, a_t = a,\, T \geq t) \\ &= \sum_{t=0}^{\infty}\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} r_t(s,a) q_t^{\boldsymbol{\pi}}(s,a)\end{aligned}$$

Theorem 4 (Sufficiency of Markov policies) *Given an MDP with state space $\mathcal{S}$ and initial state distribution μ, consider a (possibly stochastic) general policy $\boldsymbol{\pi}$. Then there exists a stochastic Markov policy $\boldsymbol{\pi}'$ such that $q_t^{\boldsymbol{\pi}} = q_t^{\boldsymbol{\pi}'}$ for all $t \geq 0$.*

Proof. For every $t \geq 0$, every state $s \in \mathcal{S}$, and every $a \in \mathcal{A}_t$ let

$$\pi'_t(a \mid s) = \frac{q_t^{\boldsymbol{\pi}}(a,s)}{\sum_{a'\in\mathcal{A}_t} q_t^{\boldsymbol{\pi}}(a',s)}$$

Clearly, $\boldsymbol{\pi}'$ is Markovian because $\pi'_t(\cdot \mid s)$ only depends on t and s. We prove that $q_t^{\boldsymbol{\pi}} = q_t^{\boldsymbol{\pi}'}$ by induction on $t \geq 0$. Let $\mathbb{P}_{\boldsymbol{\pi}}$ be the probability of states and actions when $\boldsymbol{\pi}$ is run on the MDP. For $t = 0$,

$$q_0^{\boldsymbol{\pi}'}(a,s) = \mathbb{P}_{\boldsymbol{\pi}'}\big(a_0 = a \mid s_0 = s\big)\mu(s) = \frac{q_0^{\boldsymbol{\pi}}(a,s)}{\sum_{a'\in\mathcal{A}_t} q_0^{\boldsymbol{\pi}}(a',s)}\mu(s) = q_0^{\boldsymbol{\pi}}(a,s)$$

because $\sum_{a' \in \mathcal{A}_t} q_0^{\boldsymbol{\pi}}(a', s) = \mu(s)$ by definition. Now assume $q_{t-1}^{\boldsymbol{\pi}} = q_{t-1}^{\boldsymbol{\pi}'}$ holds. Note that

$$\begin{aligned}
\mathbb{P}_{\boldsymbol{\pi}'}(s_t = s) &= \sum_{s' \in \mathcal{S}} \sum_{a' \in \mathcal{A}_t} \mathbb{P}^{\boldsymbol{\pi}'}\big(s_{t-1} = s', a_{t-1} = a', T \geq t\big) p(s \mid s', a') \\
&= \sum_{s' \in \mathcal{S}} \sum_{a' \in \mathcal{A}_t} q_{t-1}^{\boldsymbol{\pi}'}(a', s') p(s \mid s', a') \\
&= \sum_{s' \in \mathcal{S}} \sum_{a' \in \mathcal{A}_t} q_{t-1}^{\boldsymbol{\pi}}(a', s') p(s \mid s', a') && \text{(by inductive hyp.)} \\
&= \mathbb{P}_{\boldsymbol{\pi}}(s_t = s)
\end{aligned}$$

where p is the transition kernel of the MDP (which does not depend on the policy). Therefore,

$$\begin{aligned}
q_t^{\boldsymbol{\pi}'}(a, s) &= \mathbb{P}^{\boldsymbol{\pi}'}\big(a_t = a \mid s_t = s, T \geq t\big) \, \mathbb{P}^{\boldsymbol{\pi}'}\big(s_t = s, T \geq t\big) \\
&= \pi_t'(a \mid s) \, \mathbb{P}^{\boldsymbol{\pi}}\big(s_t = s, T \geq t\big) \\
&= \frac{q_t^{\boldsymbol{\pi}}(a, s)}{\sum_{a' \in \mathcal{A}_t} q_t^{\boldsymbol{\pi}}(a', s)} \mathbb{P}^{\boldsymbol{\pi}}\big(s_t = s, T \geq t\big) \\
&= q_t^{\boldsymbol{\pi}}(a, s)
\end{aligned}$$

and this concludes the proof. □

So, from now on, without loss of generality we only consider Markov policies.

For any MDP and stochastic Markov policy $\boldsymbol{\pi}$, let

$$R(\boldsymbol{\pi}) = \mathbb{E}\left[\sum_{t=0}^{T-1} \sum_{a \in \mathcal{A}_t} r_t(s_t, a) \pi_t(a \mid s_t) + r_T(s_T)\right]$$

be the return of the policy (from state s_0) computed using the stochastic horizon criterion.

Theorem 5 (Sufficiency of deterministic Markov policies) *Given an MDP with state space $\mathcal{S}$, consider a stochastic Markov policy $\boldsymbol{\pi}$. If the MDP is such that $T = t$ is independent of the trajectory of states and actions, then there exists a deterministic Markov policy $\boldsymbol{\pi}'$ such that $R(\boldsymbol{\pi}') \geq R(\boldsymbol{\pi})$ from any initial state s_0.*

Proof. Since $T = t$ does not depend on the trajectory, we can write,

$$R(\boldsymbol{\pi}) = \sum_{t=0}^{\infty} \mathbb{P}(T = t) \mathbb{E}\left[\sum_{\tau=0}^{t-1} r_\tau(s_\tau, a_\tau) + r_t(s_t)\right] = \sum_{t=0}^{\infty} \mathbb{P}(T = t) R_t(\pi_0, \ldots, \pi_t)$$

where we defined

$$R_T(\pi_0, \ldots, \pi_T) = \mathbb{E}\left[\sum_{t=0}^{T-1} r_t(s_t, a_t) + r_T(s_T)\right]$$

So, without loss of generality, we can assume T is fixed.

Given $\boldsymbol{\pi} = (\pi_0, \ldots, \pi_{T-1})$ and any $t \in [T-1]$, we prove that there exist $\pi_t', \ldots, \pi_{T-1}'$ deterministic such that $R_T(\pi_1, \ldots, \pi_{t-1}, \pi_t', \ldots, \pi_{T-1}') \geq R_T(\boldsymbol{\pi})$. The theorem is then proven for the choice $t = 0$.

The proof is by backward induction on $t \in [T-1]$. For the base case $t = T-1$, let π'_{T-1} be defined by

$$\pi'_{T-1}(s_{T-1}) = \operatorname*{argmax}_{a \in \mathcal{A}_{T-1}} \Big(r_{T-1}(s_{T-1}, a) + \mathbb{E}\big[r_T(s_T) \mid s_{T-1}, a \big] \Big)$$

Then π'_{T-1} is deterministic and

$$r_{T-1}\big(s_{T-1}, \pi'_{T-1}(s_{T-1})\big) + \mathbb{E}\big[r_T(s_T) \mid s_{T-1}, \pi'_{T-1}(s_{T-1}) \big] \geq \mathbb{E}\big[r_{T-1}(s_{T-1}, a_{T-1}) + r_T(s_T) \mid s_{T-1} \big]$$

So we have $R_T(\pi_1, \ldots, \pi_{T-2}, \pi'_{T-1}) \geq R_T(\boldsymbol{\pi})$. For the inductive step, assume there exist deterministic $\pi'_{t+1}, \ldots, \pi'_{T-1}$ such that $R_T(\pi_1, \ldots, \pi_t, \pi'_{t+1}, \ldots, \pi'_{T-1}) \geq R_T(\boldsymbol{\pi})$. Let π'_t be defined by

$$\pi'_t(s_t) = \operatorname*{argmax}_{a \in \mathcal{A}_t} \mathbb{E} \left[r_t(s_t, a) + \sum_{\tau=t+1}^{T-1} r_\tau\big(s_\tau, \pi'_\tau(s_\tau)\big) + r_T(s_T) \right]$$

Then π'_t is deterministic and

$$\begin{aligned}
&\sum_{a \in \mathcal{A}_t} \pi_t(a \mid s_t) \mathbb{E} \left[r_t(s_t, a) + \sum_{\tau=t+1}^{T-1} r_\tau\big(s_\tau, \pi'_\tau(s_\tau)\big) + r_T(s_T) \right] \\
&\quad \leq \max_{a \in \mathcal{A}_t} \mathbb{E} \left[r_t(s_t, a) + \sum_{\tau=t+1}^{T-1} r_\tau\big(s_\tau, \pi'_\tau(s_\tau)\big) + r_T(s_T) \right] \\
&\quad = \mathbb{E} \left[\sum_{\tau=t}^{T-1} r_\tau\big(s_\tau, \pi'_\tau(s_\tau)\big) + r_T(s_T) \right]
\end{aligned}$$

This shows that $R_T(\pi_1, \ldots, \pi_t, \pi'_{t+1}, \ldots, \pi'_{T-1}) \leq R_T(\pi_1, \ldots, \pi_{t-1}, \pi'_t, \ldots, \pi'_{T-1})$ concluding the proof. □