# Chapter 01 – Markov Decision Processes

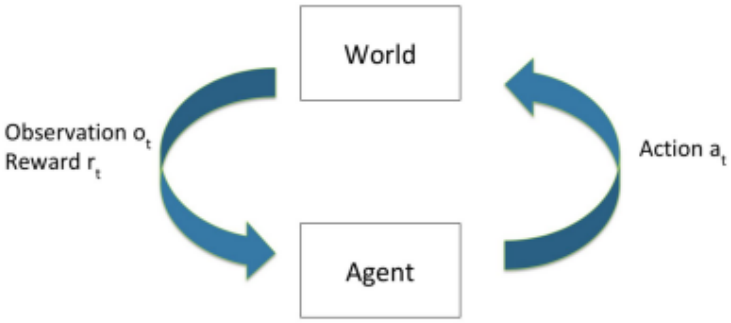
*Author: Gianmarco Scarano*

*gianmarcoscarano@gmail.com*

# 1. Sequential Decision Making

In a World, we have an Agent which interacts with the environment at discrete timesteps (we call it $t$), by receiving observations $o_t$ and reward $r_t$ from the environment after taking an action $a_t$.

Such world can be visualized here on the right.




Plus, we each interaction of the agent generates a trajectory (or history), that is used by the agent to take action:

$$h_t = (o_0, a_0, r_1, o_1, a_1, .., r_t, o_t, a_t)$$

Giving our last definition before defining the Markovian Model, we say that a state is a function of the history and it's typically hidden or unknown:

$$s_t = f(h_t)$$

# 2. Markovian Assumption & MDP

A state $s_t$ is Markovian ONLY if given the present we are in right now, the future is independent of the past.

We can write this in a formula as follows: $P(s_{t+1}|s_t, a_t) = P(s_{t+1}|h_t, a_t)$

The "$(s_t, a_t)$" part, means that I just need $(t-1, a-1)$ in order to get to the next state. I don't need history (Hence it's Markovian). I don't need any previous state. The formula above states "What is the probability of ending up in the next state, given the current state and the current action?"

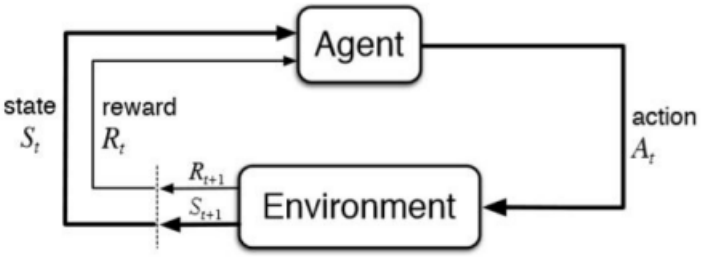
Nonetheless, a state can always be Markovian by setting it equal to the history: $s_t = h_t$

The best case is when the current state is said to be fully observable, meaning that it corresponds to the latest observation, such that $s_t = o_t$

## 2.1 Markov Decision Process (MDP)

An MDP can be described as a set of states $S = (S_0, S_1, .., S_n)$ and a set of actions $A = (a_0, a_1, .., a_n)$ which could be for example "Left / Right / Up / etc".

We can directly look at its architecture on the right, which is auto-explicative:




Here, we are looking at a Sequential Decision Making (earlier chapter) under the Markov Assumption.

### 2.1.1 Transition Function

We introduce a new concept, which is really important, called TRANSITION FUNCTION ($T$), which is the probability of the next state given the current state and the current action. Generally, $T$ is stochastic and not deterministic.

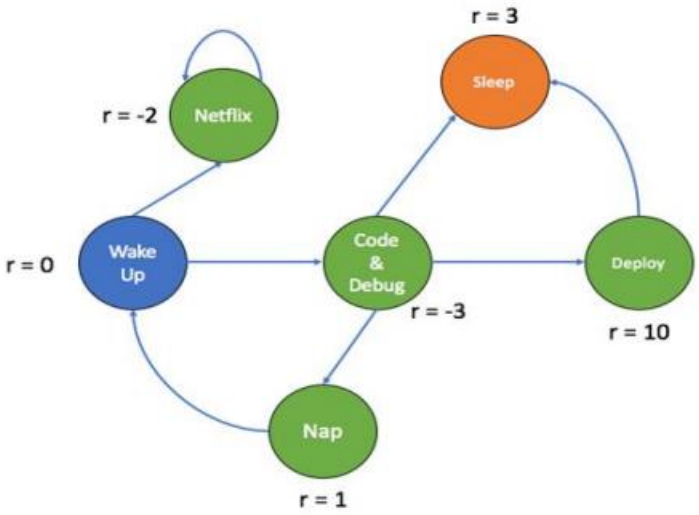
### 2.1.2 Reward

Generally, a reward $r_t$ is a number/scalar representing a feedback and indicates how well an agent is doing at step $t$.

We indicate the reward as: $R(s, a)$.

Cost is the inverse of the reward.

### 2.1.3 Deterministic MDP



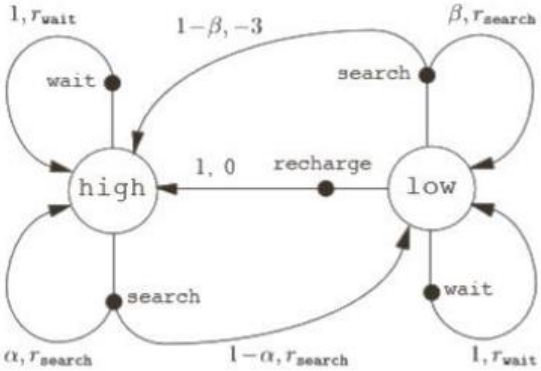

When we deal with a deterministic approach, the probability of getting to the next state is always equal to 1. There is no way we can say that ending up in a state $s$ has a higher probability than another state $s'$.

This example is called episodic, due to the fact that it actually finishes (orange circle).

### 2.1.4 Stochastic MDP

Recycling robot

| $s$ | $a$ | $s'$ | $p(s' \mid s, a)$ | $r(s, a, s')$ |
|---|---|---|---|---|
| high | search | high | $\alpha$ | $r_{\text{search}}$ |
| high | search | low | $1-\alpha$ | $r_{\text{search}}$ |
| low | search | high | $1-\beta$ | $-3$ |
| low | search | low | $\beta$ | $r_{\text{search}}$ |
| high | wait | high | 1 | $r_{\text{wait}}$ |
| high | wait | low | 0 | - |
| low | wait | high | 0 | - |
| low | wait | low | 1 | $r_{\text{wait}}$ |
| low | recharge | high | 1 | 0 |
| low | recharge | low | 0 | - |



# 3. Policy, Infinite Horizon, Value/Q-Function, etc.

In this chapter, we'll the discuss the basics of Reinforcement Learning algorithms.

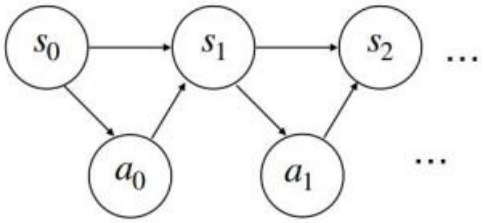
## 3.1 Policy

A policy $\pi$ maps ALL the states $\{S\}$ into actions $\{A\}$ and determines how agents select actions.

It can be deterministic, meaning that an action is described as follows: $a = \pi(s)$ or it can be stochastic, meaning that there is uncertainty for the action the agent is going to take: $\pi(a|s)$ -> What is the probability of executing a certain action, given a certain state, following a certain policy $\pi$?

## 3.2 Trajectory Probability

Here the question is, what is the probability of seeing a trajectory at time $t$ according to a policy $\pi$ starting from $s_0$?




$$\mathbb{P}^{\pi}(s_0, a_0, \ldots s_t, a_t) = \pi(a_0|s_0)p(s_1|s_0, a_0)\pi(a_1|s_1)p(s_2|s_1, a_1)\ldots p(s_t|s_{t-1}, a_{t-1})\pi(a_t|s_t)$$

The $(s_0, a_0, \ldots\ , s_t, a_t)$ part just after $P^{\pi}$ is the trajectory at time t starting from $s_0$. The $\pi$ symbol at the top of P, means that we are following a certain policy.

Another way to compact this is to ask ourselves what is the probability of visiting a certain state $(s, a)$ at time $t$ according to a certain policy $\pi$ and starting from $s_0$?

We can easily sum it up and compress it to a simple formula which converts this question in a mathematical form:

$$\mathbb{P}^{\pi}{}_t(s, a; s_0) = \sum_{a0, s1, a1, \ldots st-1, at-1} \mathbb{P}^{\pi}(s_0, a_0, \ldots s_t = s, a_t = a)$$

## 3.3 Infinite Horizon

So far, in our MDP we have $(S, A, T, R)$ -> State, Action, Transition Function, Reward

We have to think and reason about policy's long term-effects.

For doing this, we introduce a discount factor $\gamma$ which is in range [0, 1).

$\gamma = 0$ means that I only care about immediate rewards.

$\gamma = 1$ means that immediate and future rewards are equally important.

1 is excluded due to the fact that, being $\sum_{h=0}^{\infty} y^h$ (which we'll see in the next subchapter) a geometric series, it is equivalent to $\frac{1}{(1-y)}$ and so setting $\gamma = 1$ for infinite tasks is a bad idea.

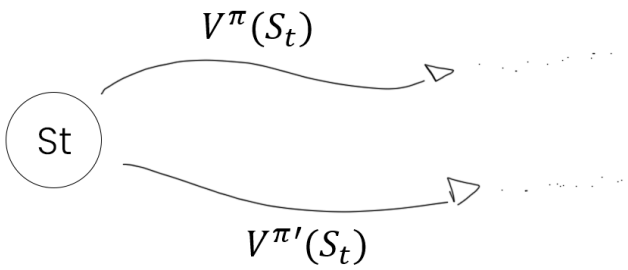
## 3.4 Value Function / Q-Function

The Value Function estimates the goodness of states and actions based on their values. It's also a nice way to compare different policies.

$$V^{\pi}(s_t) = \mathbb{E}_{\pi}[r_t+\gamma r_{t+1}+\gamma^2 r_{t+2}+\gamma^3 r_{t+3}+\ldots|s_t] = \mathbb{E}[\sum_{h=0}^{\infty}\gamma^h r_h|s_0=s_t, a_h=\pi(s_h), s_{h+1}\sim p(.|s_h,a_h)]$$

Here, we can see that we compute the Expectation ($\mathbb{E}_{\pi}$) of each reward discounted by the reward gamma ($\gamma$). The expectation is more or less a weighted average of all possible probabilities. In this case probability of getting a reward at time $(t+1, t+2 \ldots)$ given a state S ($s_t$).

In this case if $\gamma = 0$, then we care only about current reward $r_t$ (since $r_{t+1}, r_{t+2}, r_{t+3}$ etc. will be 0)




Here, we give the definition of Q-Function instead:

$$Q^{\pi}(s_t,a_t) = \mathbb{E}[\sum_{h=0}^{\infty}\gamma^h r_h|(s_0,a_0)=(s_t,a_t), a_{h+1}=\pi(s_h), s_{h+1}\sim p(.|s_h,a_h)]$$

The main difference from the Value function, is the following:

- Q Function:
  - I take each possible action (IN THE CURRENT STATE) and ONLY starting from NEXT STATE, I will follow my policy $\pi$.
- Value Function:
  - From the CURRENT STATE, I take an action "$a$" according to my policy and afterwards I will follow my policy $\pi$.

So, in the Q-Function what we do is that we have the flexibility of choosing any possible action and then following my policy. This is a way to compute which action leads me to the best reward.

While, in the Value function this doesn't happen, because we take only 1 action $(a)$ and after that we'll follow the policy, so I'm not aware of what could happen if I choose any other action.

**EXAMPLES**:

$S =$

| | | | |
|---|---|---|---|
| -0.1 | -0.1 | -0.1 | +10 (G) |
| -0.1 | -0.1 | -0.1 | -0.1 |
| 0 | -0.1 | -0.1 | -0.1 |

$\pi_1 =$

| | | | |
|---|---|---|---|
| ↑ | ↑ | ↑ | ↑ |
| ↑ | ↑ | ↑ | ↑ |
| ↑ | ↑ | ↑ | ↑ |

$\pi_2 =$

| | | | |
|---|---|---|---|
| → | → | → | ← |
| ↑ | → | → | ↑ |
| → | ↑ | → | ↑ |

$\gamma = 0.9$

$A = \{\rightarrow\leftarrow\uparrow\downarrow\}$

---

$V^{\pi_1}(S_{0,0}) = 0 + 0.9 * (-0.1) + 0.9^2 * (-0.1) + 0.9^3 * (-0.1) + 0.9^4 * (-0.1) + 0.9^5 * (-0.1) + 0.9^6 * (-0.1) = -0.569$

We stopped at $0.9^6$ because we fixed the iteration. We should be doing this up to infinity or up to a fixed term.

---

$V^{\pi_2}(S_{0,0}) = 0 + 0.9 * (-0.1) + 0.9^2 * (-0.1) + 0.9^3 * (-0.1) + 0.9^4 * (-0.1) + 0.9^5 * 10 + 0.9^6 * (-0.1) = 5.54$

From here I already know something.

---

$Q^{\pi_2}(S_{0,0}, \uparrow) = 0 + 0.9 * (-0.1) + 0.9^2 * (-0.1) + 0.9^3 * (-0.1) + 0.9^4 * (-0.1) + 0.9^5 * 10 + 0.9^6 * (-0.1) = 5.54$

Here what I do is that in state 0,0 I replace the policy action ($\rightarrow$) with the action that I specified in the arguments: ↑

---

$Q^{\pi_2}(S_{3,2}, \uparrow) = 10 + 0.9 * 10 + 0.9^2 * (-0.1) + 0.9^3 * 10 + 0.9^4 * (-0.1) + 0.9^5 * 10 + 0.9^6 * (-0.1) = 31.995$

Here what I do is that I start in state 3,2 and I immediately replace the policy action ($\leftarrow$) with the action that I specified in the arguments: ↑

---

$V^{\pi_2}(S_{3,2}) = 10 + 0.9 * (-0.1) + 0.9^2 * 10 + 0.9^3 * (-0.1) + 0.9^4 * 10 + 0.9^5 * (-0.1) + 0.9^6 * 10 = 29.75$

## 4. Bellman Equation

In the Bellman Equation, a value of a certain state is expanded in terms of the current reward and the value of the next states according to the policy.

r here is function of s and $\pi(s)$

$$V^{\pi}(s_t) = \mathbb{E}_{\pi}[r_t+\gamma r_{t+1}+\gamma^2 r_{t+2}+\gamma^3 r_{t+3}+\ldots|s_t] = r_t+\gamma \mathbb{E}_{s'\sim p(.|s,\pi(s))}[V^{\pi}(s')]$$

$$Q^{\pi}(s_t,a) = r_t+\gamma \mathbb{E}_{s'\sim p(.|s,a)}[V^{\pi}(s')]$$

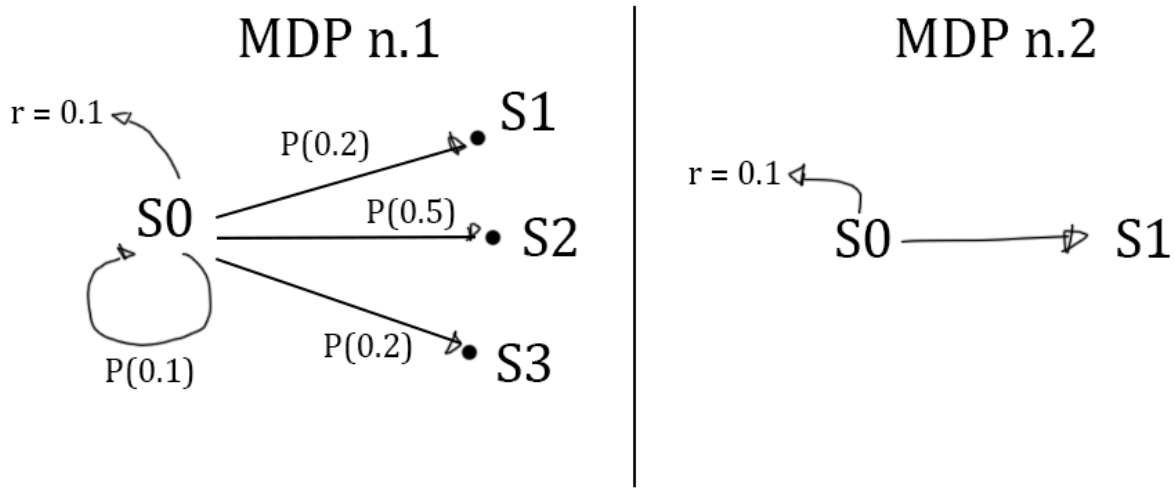
r here is function of s and a

As a result V(s) = Q(s,$\pi$(s))

We can see that the Value function of $S_t$ is the Reward at the initial state $(r_t)$ + $\gamma$ which multiplies the Expectation with respect to the transition function (so according to the policy!). That's why it's written that "r here is a function of S and $\pi(s)$".

In the Q function instead, the Expectation lives w.r.t to a certain action "a". For this reason, if I use the Q-Function with the policy, I'll end up in the Value function.

**EXAMPLE:**




Let's consider this situation with $\gamma = 0.9$ and the following V-table (which is given):

$S_0 = 2\,;\ S_1 = 10\,;\ S_2 = 2\,;\ S_3 = -10$

Then:

$$V(S_0)_{MDP\_2} = 0.1\ (reward) + 0.9\ (\gamma) * 10\ (which\ comes\ from\ the\ V-table) = 9.1$$

$$V(S_0)_{MDP_1} = 0.1\ (reward) + 0.9\ (\gamma) * [0.1(prob\ of\ S_0) *\ 2\ (V\ of\ S_0) + 0.2\ (prob\ of\ S_1) * 10\ (V\ of\ S_1) + 0.5 * 2 + 0.2 * (-10) = 1.18$$

## 4.1 Optimal Policy

Very simply, optimal policy states that for infinite horizon MDPs, there always exists a deterministic policy $\pi^*$ such that:

$$V^{\pi^*}(s) \geq V^{\pi}(s) \; \forall \; s, \pi$$

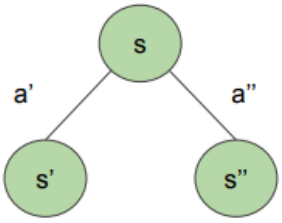
Meaning that $\pi^*$ dominates all other policies $\pi$ in each state.

# 5. Bellman Optimality

$$V^*(s)=\max_a [r(s,a)+\gamma \mathbb{E}_{s' \sim p(.|s,a)} V^*(s')]$$

$Q^*(s,a)$

- Try a', get r(s,a'), compute $Q^*(s,a')=r(s,a')+\gamma V^*(s')$
- Try a'', get r(s,a''), compute $Q^*(s,a'')=r(s,a'')+\gamma V^*(s'')$




Assume we know $V^*$ at s' and s''

$$V^*(s)=\max_{a',a''}\{Q^*(s,a'),Q^*(s,a'')\}$$

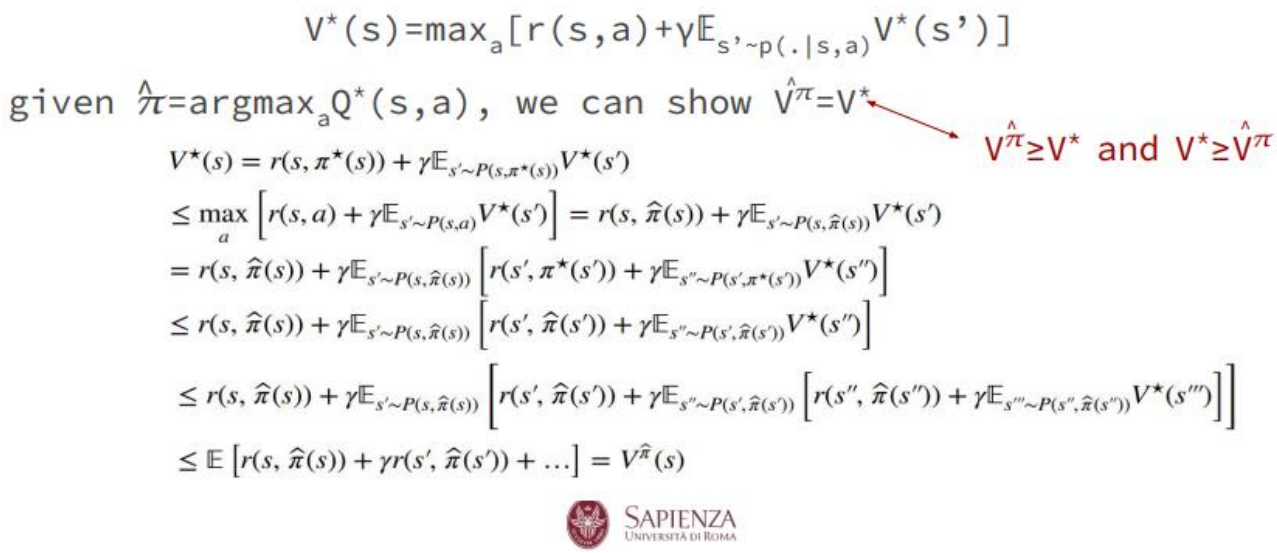
Commenting this picture, we see that $V^*$is equal to choosing the best first action among the twos which maximizes that whole quantity at that state ($Q^*(s,a)$) and the difference with the Bellman Equation of subchapter 4 it's literally this.

However, though, one assumes that we know which is the optimal Value function in the next state.

As we can see from the bullet points, we compute the Q-function trying all actions (a' and a''), then we choose the best one.

## 5.1 Bellman Optimality (Theorem 1)

This theorem states that given $\hat{\pi}$ = $argmax_a\ Q^*(s,a)$, we can show that $V^{\hat{\pi}} = V^*$




Commenting this picture as well, if I know the argument of the max function of $V^*(s)$ -> $r(s,a) + \gamma\ etc.$ (which is essentially Q*(s,a)) -> and I take the action that maximizes the reward right now, what we get is the optimal policy.

Now, given that $\hat{\pi}$ is the policy that chooses the action which maximizes my Q-function I want to prove that $V^*(s) \leq V^{\hat{\pi}}(s)$.

For accomplishing this, we see in the 2nd row that the max operation + its argument gets directly replaced by the quantity on the right -> $r(s, \hat{\pi}(s)\ etc.)$. This is due to the fact that we know that $\hat{\pi}$ is that one policy which maximizes the Q-function (which is written just below the sub-chapter 5). That's why instead of writing $(r(s,a)\ etc.)$ we write $r(s, \hat{\pi}(s)\ etc.)$ -> Because the maximization is given by the policy $\hat{\pi}$.

Now, very simply if we keep on expanding this for all the states and weighting it through the expectation, we prove that $V^*(s) = V^{\hat{\pi}}(s)$.

## 5.2 Bellman Optimality (Theorem 2)

For any V, if $V(s)=\max_a[r(s,a)+\gamma\mathbb{E}_{s'\sim p(.|s,a)}V(s')]$ for all s, then $V(s)=V^*(s)$

We need to check if $|V(s)-V^\star(s)| = \left|\max_a(r(s,a)+\gamma\mathbb{E}_{s\sim P(s,a)}V(s')) - \max_a(r(s,a)+\gamma\mathbb{E}_{s\sim P(s,a)}V^\star(s'))\right|$

$$\leq \max_a \left|(r(s,a)+\gamma\mathbb{E}_{s\sim P(s,a)}V(s')) - (r(s,a)+\gamma\mathbb{E}_{s\sim P(s,a)}V^\star(s'))\right|$$

$$\leq \max_a \gamma\mathbb{E}_{s'\sim P(s,a)}\left|V(s')-V^\star(s')\right|$$

$$\leq \max_a \gamma\mathbb{E}_{s'\sim P(s,a)}\left(\max_{a'}\gamma\mathbb{E}_{s''\sim P(s',a')}\left|V(s'')-V^\star(s'')\right|\right)$$

At infinity, this goes to zero

$$\leq \max_{a_1,a_2,\ldots a_{k-1}} \gamma^k\mathbb{E}_{s_k}|V(s_k)-V^\star(s_k)|$$

SAPIENZA UNIVERSITÀ DI ROMA

By commenting this, in order to check if $V(s) = V^*(s)$, there is just one way: the difference of the norm is equal to 0 -> $|V(s) - V^*(s)| = 0$

We simply expand both $V(s)$ and $V^*(s)$ at the first step.

At the second step, by using a simple property of the norms we take out the max operation and cancel out $r(s,a)$.

At the third step, I take out $\gamma$ which multiplies the expectation (since they are equal for both terms).

At the very end, I do the maximum of all the possible actions and since $\gamma \in [0,1]$ gets elevated to $k$ (which is the number of times I did this iteration), this goes to zero (since we are in an Infinite Horizon setting), proving that the difference between $|V(s) - V^*(s)|$ is indeed 0.