Mahindra™ University
Global Thinkers. Engaged Leaders.

Reinforcement Learning and Autonomous Systems (CS4122)

Lecture 14 (16/09/2024)
Lecture 15 (18/09/2024)
Lecture 16 (23/09/2024)

Instructor: Gourav Saha

Lecture Content

- Setup of Markov Decision Process.
- Example of Markov Decision Process.
- Episodic vs Continuing Tasks.

Recap of Module 2 (covered till now)

- **Lecture 10:** We defined the components of a Markov Decision Process (MDP). The components are:
  - State and state space.
  - Action and action space.
  - Reward and reward probability.
  - State transition probability.

- **Lectures 11-13:** Formulating (real-world) decision making problems as MDP.
  - Fishing in gridworld (lecture 10).
  - Automatic Scheduler for WiFi and Mobile Data (lecture 11).
  - Investing in Stock Market (lectures 12 and 13).

Now what?

Recap of Module 2 (covered till now)

- **Lecture 10:** We defined the **components of a Markov Decision Process** (MDP). The components are:
  - State and state space.
  - Action and action space.
  - Reward and reward probability.
  - State transition probability.

- **Lectures 11-13:** **Formulating (real-world) decision making problems as MDP**.
  - Fishing in gridworld (lecture 10).
  - Automatic Scheduler for WiFi and Mobile Data (lecture 11).
  - Investing in Stock Market (lectures 12 and 13).

Now what?

Answer: Find optimal policies for MDP. This is what we will do for the remaining lectures of Module 2.

What do we want to optimize?

- Recall (from lectures 2 and 3) that for MDPs, the policy is a mapping from state, x, to the probability of all the actions $a \in \mathcal{A}(x)$,

$$\boldsymbol{\pi}(\boldsymbol{a} \mid \boldsymbol{x}) \quad \text{(E.1.)}$$

NOTE: **(E.1.) is a randomized policy**. A policy can even be deterministic. That said, a **deterministic policy is a special case of randomized policy**. Hence, by learning about randomized policy, we are also learning about deterministic policies also.

What do we want to optimize?

- Recall (from lectures 2 and 3) that for MDPs, the policy is a mapping from state, x, to the probability of all the actions $a \in \mathcal{A}(x)$,

$$\boldsymbol{\pi}(\boldsymbol{a} \mid \boldsymbol{x}) \tag{E.1.}$$

NOTE: **(E.1.) is a randomized policy**. A policy can even be deterministic. That said, a **deterministic policy is a special case of randomized policy**. Hence, by learning about randomized policy, we are also learning about deterministic policies also.

- We want to find the optimal policy, π. But, before we can even talk about finding the optimal policy, we wave to answer the following question:

What do we want to optimize? Or mathematically, **what is the objective function**?

Let's answer this question first.

What do we want to optimize?

There can be three possible objection functions in MDPs.

Objective Function 1
(Sum of rewards)

$$E_\pi\left[\sum_{t=0}^{T} r_t \,\middle|\, x_0 = x\right]$$

Only when time horizon T is finite, i.e. episodic task.

Objective Function 2
(Discounted reward)

$$E_\pi\left[\sum_{t=0}^{\infty} \beta^t r_t \,\middle|\, x_0 = x\right]$$

Suitable when time horizon T is infinite, i.e. continuing task.

Discounting factor, $\beta \in [0,1)$.

Objective Function 3
(Average reward)

$$\lim_{T\to\infty}\frac{1}{T+1}E_\pi\left[\sum_{t=0}^{T} r_t \,\middle|\, x_0 = x\right]$$

Suitable when time horizon T is infinite, i.e. continuing task.

What do we want to optimize?

There can be three possible objection functions in MDPs.

Objective Function 1
(Sum of rewards)

$$E_{\pi}\left[\sum_{t=0}^{T} r_t \,\middle|\, x_0 = x\right]$$

Objective Function 2
(Discounted reward)

$$E_{\pi}\left[\sum_{t=0}^{\infty} \beta^t r_t \,\middle|\, x_0 = x\right]$$

Objective Function 3
(Average reward)

$$\lim_{T\to\infty} \frac{1}{T+1} E_{\pi}\left[\sum_{t=0}^{T} r_t \,\middle|\, x_0 = x\right]$$

- The **decision variable** of the optimization problem is the policy, π.

What do we want to optimize?

There can be three possible objection functions in MDPs.

Objective Function 1
(Sum of rewards)

$$E_\pi\left[\sum_{t=0}^{T} r_t \,\middle|\, x_0 = x\right]$$

Objective Function 2
(Discounted reward)

$$E_\pi\left[\sum_{t=0}^{\infty} \beta^t r_t \,\middle|\, x_0 = x\right]$$

Objective Function 3
(Average reward)

$$\lim_{T\to\infty} \frac{1}{T+1} E_\pi\left[\sum_{t=0}^{T} r_t \,\middle|\, x_0 = x\right]$$

- The **decision variable** of the optimization problem is the policy, π.
- This term means that the **initial state is $\boldsymbol{x}$**.

What do we want to optimize?

There can be three possible objection functions in MDPs.

Objective Function 1
(Sum of rewards)

$$E_\pi \left[\sum_{t=0}^{T} r_t \,\middle|\, x_0 = x \right]$$

Objective Function 2
(Discounted reward)

$$E_\pi \left[\sum_{t=0}^{\infty} \beta^t r_t \,\middle|\, x_0 = x \right]$$

Objective Function 3
(Average reward)

$$\lim_{T\to\infty} \frac{1}{T+1} E_\pi \left[\sum_{t=0}^{T} r_t \,\middle|\, x_0 = x \right]$$

- It can be shown that under some "nice conditions", we can find the optimal policy for objective 3 (average reward) by simply finding the optimal policy for objective 2 (discounted reward) but **letting $\boldsymbol{\beta} \to \mathbf{1}$** (in practice we will choose β very close to 1; say **0.99**). This is called **"Blackwell optimality"** [1].

- So, we can **ignore objective** 3 and focus on objective 2.

[1] Dimitri Bertsekas, "Dynamic Programming and Optimal Control", Vol. 2, 3rd ed, Chapter 4.

What do we want to optimize?

There can be three possible objection functions in MDPs.

Objective Function 1
(Sum of rewards)

$$E_\pi \left[\sum_{t=0}^{T} r_t \,\middle|\, x_0 = x \right]$$

Objective Function 2
(Discounted reward)

$$E_\pi \left[\sum_{t=0}^{\infty} \beta^t r_t \,\middle|\, x_0 = x \right]$$

- Now recall that in lecture 10 slides we discussed that we can **convert an episodic task in continuing task** by introducing a **"end state"**.
- If we do the above, objective function 1 which is meant for episodic task, is **approximately equal** to objective function 2 if we let $\boldsymbol{\beta \to 1}$.
- So, we can **ignore objective 1** and focus on objective 2.

What do we want to optimize?

There can be three possible objection functions in MDPs.

Objective Function 2
(Discounted reward)

$$E_\pi\left[\sum_{t=0}^{\infty} \beta^t r_t \,\middle|\, x_0 = x\right]$$

- Finally, the objective function that we will concentrate on for the remaining modules is the **expected value** of the **discounted reward under policy π**.

- We will evaluate the other two objective functions also. But we will develop algorithms to compute the optimal policy for discounted reward only. This is similar to how in ML/DL we minimize loss function (analogous to discounted reward) even though the actual metric (analogous to sum of reward and average reward) that we are interested in something different.

Roadmap to Compute Optimal Policy

- Now that we have decided what is the objective function, we have to find an approach to optimize this objective function and hence find the optimal policy. The roadmap to find the optimal policy is as follows:

Step 1: Policy evaluation

Step 1.1: Iterative policy evaluation

Step 1.2: Bellman equation

In order to improve a policy, we first need to identify a way to evaluate its performance. That's what policy evaluation does; i.e. **policy evaluation** means to evaluate the value of the objective function given a policy. The solution of the **Bellman equation** is measure of how good a policy is performing and **iterative policy evaluation** is an approach to solve the Bellman equation.

Roadmap to Compute Optimal Policy

- Now that we have decided what is the objective function, we have to find an approach to optimize this objective function and hence find the optimal policy. The roadmap to find the optimal policy is as follows:

Step 1: Policy evaluation

Step 1.1: Iterative policy evaluation

Step 1.2: Bellman equation

Step 2: Computing Optimal Policy

Step 2.1: **Value iteration**

Step 2.2: Bellman optimality equation

Roadmap to Compute Optimal Policy

➢ Now that we have decided what is the objective function, we have to find an approach to optimize this objective function and hence find the optimal policy. The roadmap to find the optimal policy is as follows:

Dynamic Programming Approach

Step 1: Policy evaluation

- **Step 1.1:** Iterative policy evaluation
- **Step 1.2:** Bellman equation

Step 2: Computing Optimal Policy

- **Step 2.1:** Value iteration
- **Step 2.2:** Bellman optimality equation

Step 3: Formal approach to derive Bellman equation and Bellman optimality equation

Step 4: Policy Iteration

Policy Improvement Theorem

Step 5: An alternative viewpoint of Value Iteration

Step 6: Generalized Policy Iteration

Policy Evaluation

- Policy evaluation means to compute the value of the objective function,

$$E_\pi\left[\sum_{t=0}^{\infty} \beta^t r_t \,\middle|\, x_0 = x\right]$$

under policy π for all values of the initial state x.

NOTE: In general, the term policy evaluation is used for other objective functions (like sum of rewards and average rewards also. But in this course, policy evaluation by default will be used for discounted reward unless mentioned otherwise.)

Policy Evaluation

- Policy evaluation means to compute the value of the objective function,

$$E_\pi\left[\sum_{t=0}^{\infty}\beta^t r_t \,\middle|\, x_0 = x\right]$$

under policy π for all values of the initial state x. In order to compute the above, we need to define a few quantities.

Definition (Return): Return, G_t, is the net discounted reward starting from time t,

$$\begin{aligned} G_t &= r_t + \beta r_{t+1} + \beta^2 r_{t+2} + \cdots \\ &= \sum_{k=0}^{\infty} \beta^k r_{t+k} \end{aligned}$$

NOTE: Return G_t is same as the discounted reward here with the only difference that return G_t is defined for any starting time t rather than only for $t = 0$. So, return is a more general version of the discounted reward from time $t = 0$.

Policy Evaluation

- Policy evaluation means to compute the value of the objective function,

$$E_\pi\left[\sum_{t=0}^{\infty}\beta^t r_t \,\middle|\, x_0 = x\right]$$

  under policy π for all values of the initial state x. In order to compute the above, we need to define a few quantities.

<u>**Definition (Value function)**</u>**:** The value function of state, x, under policy, π, is the expected value of the return, G_t, given that $x_t = x$ (i.e. starting from state x at time t) and the agent uses policy π to decide its action at all time,

$$V^\pi(x) = E_\pi\left[G_t \mid x_t = x\right]$$

NOTE: Two things to note:

(i) Value function is same the objective function with the only difference that the starting time t can vary for value function. Hence, value function is more general than the objective function.

Policy Evaluation

- Policy evaluation means to compute the value of the objective function,

$$E_\pi\left[\sum_{t=0}^{\infty}\beta^t r_t \,\middle|\, x_0 = x\right]$$

under policy π for all values of the initial state x. In order to compute the above, we need to define a few quantities.

Definition (Value function): The value function of state, x, under policy, π, is the expected value of the return, G_t, given that $x_t = x$ (i.e. starting from state x at time t) and the agent uses policy π to decide its action at all time,

$$V^\pi(x) = E_\pi\left[G_t \mid x_t = x\right]$$

NOTE: Two things to note:
(ii) Value function is not a function of starting time t (no t in the argument). This is because for any two times t_1 and t_2, G_{t_1} and G_{t_2} are time-shifted (discounted) sum of an infinite series of random variable r_t. Since the involved **probability distributions** (state transitions probability, reward probability, and the policy) **are stationary**, a time shift will NOT change the expected value of a sum of an **infinite series**.

Policy Evaluation

- Policy evaluation means to compute the value of the objective function,

$$E_\pi\left[\sum_{t=0}^{\infty} \beta^t r_t \,\middle|\, x_0 = x\right]$$

under policy π for all values of the initial state x. In order to compute the above, we need to define a few quantities.

Definition (Action-Value function): The action-value function of state action pair, (x, a), under policy, π, is the expected value of the return, G_t, given that $x_t = x$ (i.e. starting from state x at time t) and $a_t = a$, and the agent uses policy π to decide its action at all time but t,

$$q^\pi(x, a) = E_\pi\left[G_t \mid x_t = x, a_t = a\right]$$

Action-value function is often called the Q-function.

NOTE: Two things to note:

(i) The only difference between value function and action-value function is that for action-value function, the agent uses any action a, not necessarily generated by policy π, at starting time t.

(ii) Action-value function does not change with time starting time t. Same explanation as value function.

Policy Evaluation

- Return G_t, and correspondingly value function and action-value function are defined for **infinite optimization horizon**. We now define a **finite horizon** equivalent of return, value function, and action-value function.

Definition (Return for finite optimization horizon): Return, $G_{k,T}$, for optimization horizon, T, is the net discounted reward in the last k time slot of the optimization horizon,

$$\begin{aligned} G_{k,T} &= r_{T-k+1} + \beta r_{T-k+2} + \beta^2 r_{T-k+3} + \cdots + \beta^{k-1} r_T \\ &= \sum_{j=0}^{k-1} \beta^j r_{T-k+1+j} \end{aligned}$$

Policy Evaluation

- Return G_t, and correspondingly value function and action-value function are defined for infinite optimization horizon. We now define a finite horizon equivalent of return, value function, and action-value function.

<u>**Definition (Value function for finite optimization horizon)**</u>**:** The value function for finite optimization horizon corresponding to state, x, under policy, π, is the expected value of the return, $G_{k,T}$, given that $x_{T-k+1} = x$ (i.e. starting from state x at time $T-k+1$) and the agent uses policy π to decide its action at all time,

$$V_{k,T}^{\pi}(x) = E_\pi\left[G_{k,T} \mid x_{T-k+1} = x\right]$$

<u>**Definition (Action-Value function for finite optimization horizon)**</u>**:** The action-value function for finite optimization horizon corresponding to state action pair, (x, a), under policy, π, is the expected value of the return, $G_{k,T}$, given that $x_{T-k+1} = x$ (i.e. starting from state x at time $T - k + 1$) and $a_{T-k+1} = a$, and the agent uses policy π to decide its action at all time but $T - k + 1$,

$$q_{k,T}^{\pi}(x) = E_\pi\left[G_{k,T} \mid x_{T-k+1} = x, a_{T-k+1} = a\right]$$

Policy Evaluation

- Return G_t, and correspondingly value function and action-value function are defined for infinite optimization horizon. We now define a finite horizon equivalent of return, value function, and action-value function.

Unlike infinite horizon, for finite horizon, the value and the action value functions depend on starting time $T - k + 1$.

Definition (Value function for finite optimization horizon): The value function for finite optimization horizon corresponding to state, x, under policy, π, is the expected value of the return, $G_{k,T}$, given that $x_{T-k+1} = x$ (i.e. starting from state x at time $T-k+1$) and the agent uses policy π to decide its action at all time,

$$V^{\pi}_{k,T}(x) = E_{\pi}\left[G_{k,T} \mid x_{T-k+1} = x\right]$$

Definition (Action-Value function for finite optimization horizon): The action-value function for finite optimization horizon corresponding to state action pair, (x, a), under policy, π, is the expected value of the return, $G_{k,T}$, given that $x_{T-k+1} = x$ (i.e. starting from state x at time $T-k+1$) and $a_{T-k+1} = a$, and the agent uses policy π to decide its action at all time but $T-k+1$,

$$q^{\pi}_{k,T}(x) = E_{\pi}\left[G_{k,T} \mid x_{T-k+1} = x, a_{T-k+1} = a\right]$$

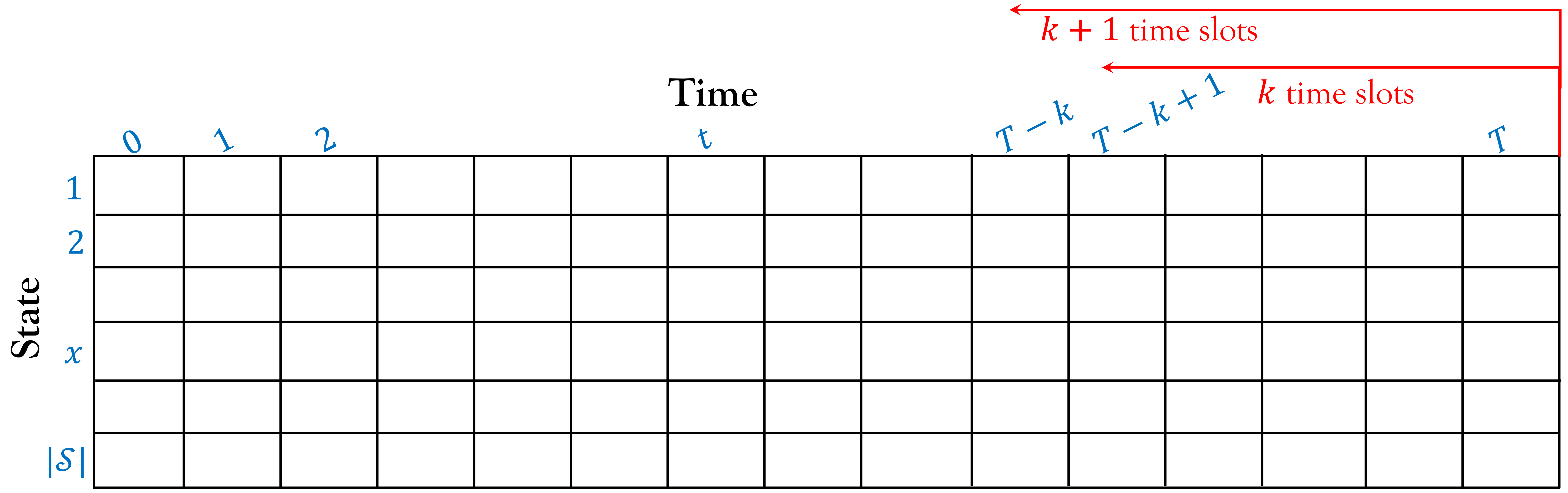
Policy Evaluation

In this (click here) slide, we discussed that the value function, $V^{\pi}(x)$, is same as the objective function. In what follows, we do the following:

- First, we derive an expression to compute the value function, $V^{\pi}_{k,T}(x)$, for the finite horizon case. This expression that we derive is called **iterative policy evaluation (IPE)**.
- Second, we then use the expression for $V^{\pi}_{k,T}(x)$ to derive an expression for $V^{\pi}(x)$. This expression is called the **Bellman equation**.

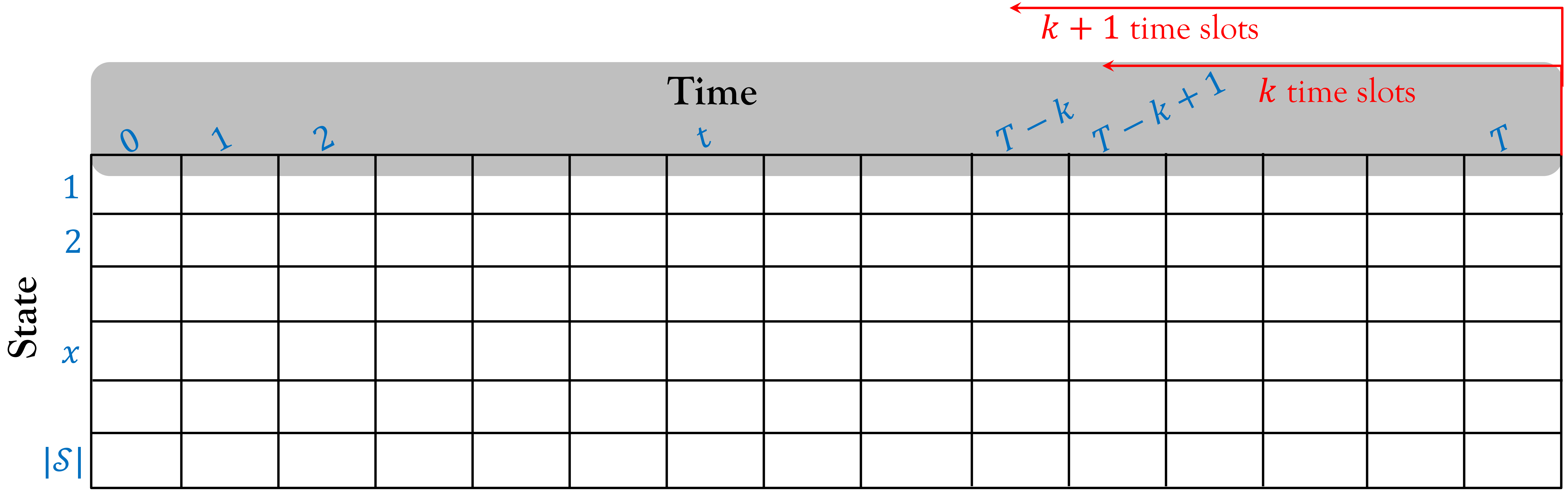
NOTE: The derivation of IPE and Bellman equation that we do next is suitable for those who are comfortable with Dynamic programming. It is good for developing intuition. A more rigorous derivation of IPE and Bellman equation is discussed later in this (click here) slide.

Policy Evaluation: IPE and Bellman Equation



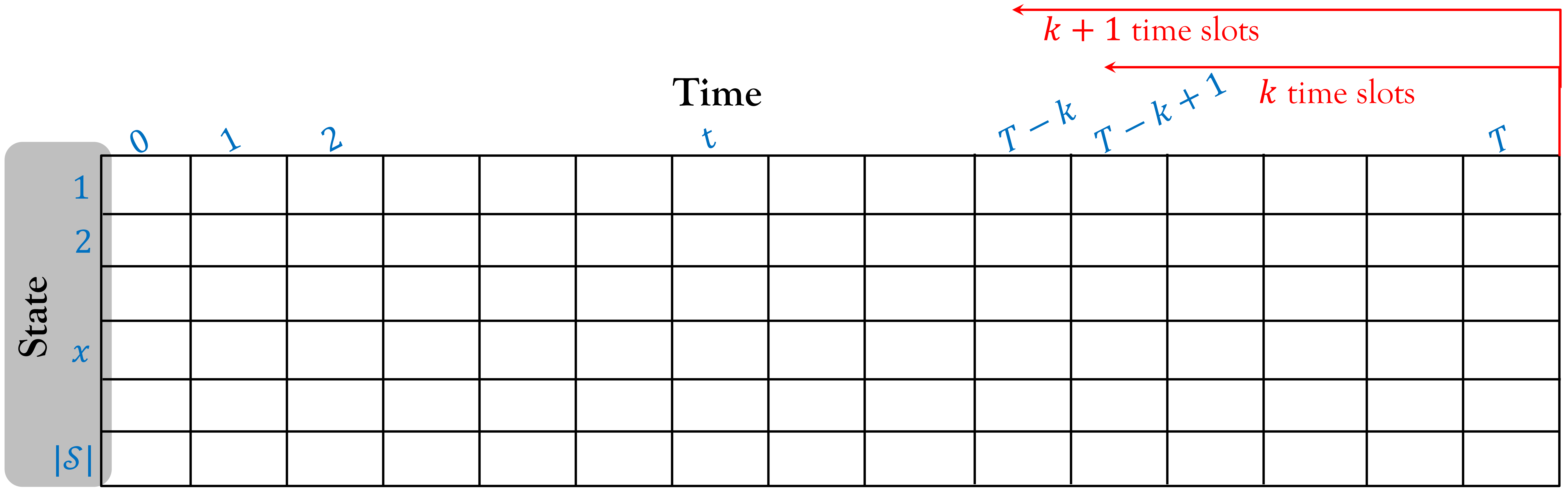
- As mentioned in the previous slide, we will use dynamic programming (DP) to find an expression for $V_{k,T}^{\pi}(x)$. If you recall, for many of the dynamic programming problem that you might have solved, you had to fill a table. Keeping up with the same spirit, a table is shown above.

Policy Evaluation: IPE and Bellman Equation



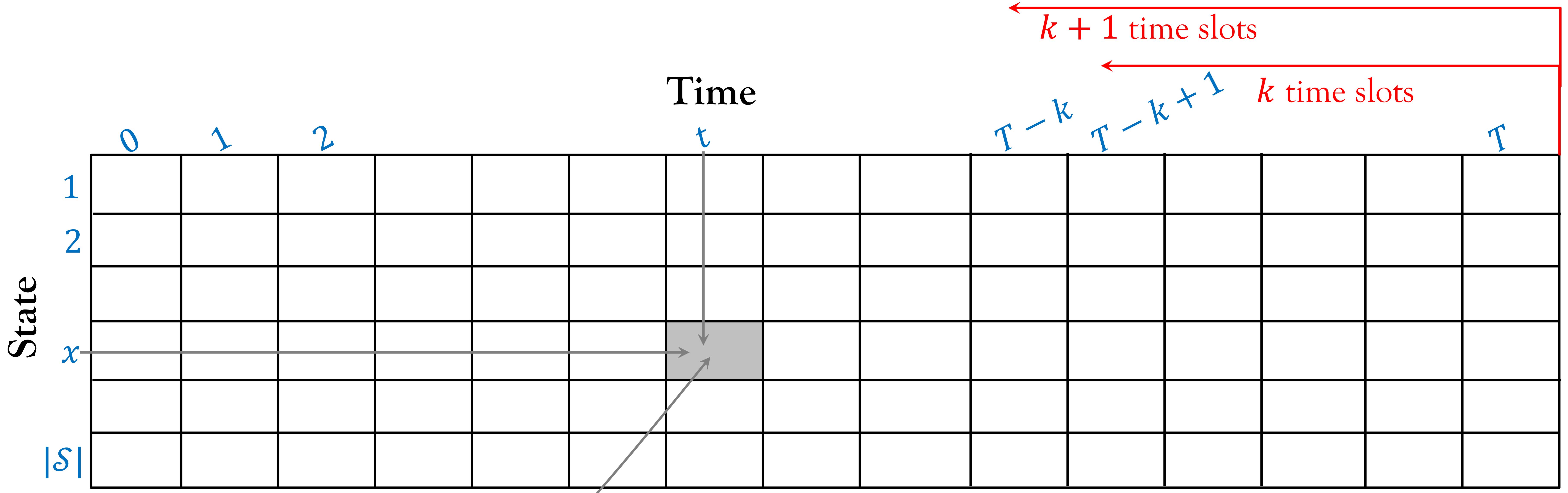
- Each column of the table is associated with a time t.

Policy Evaluation: IPE and Bellman Equation



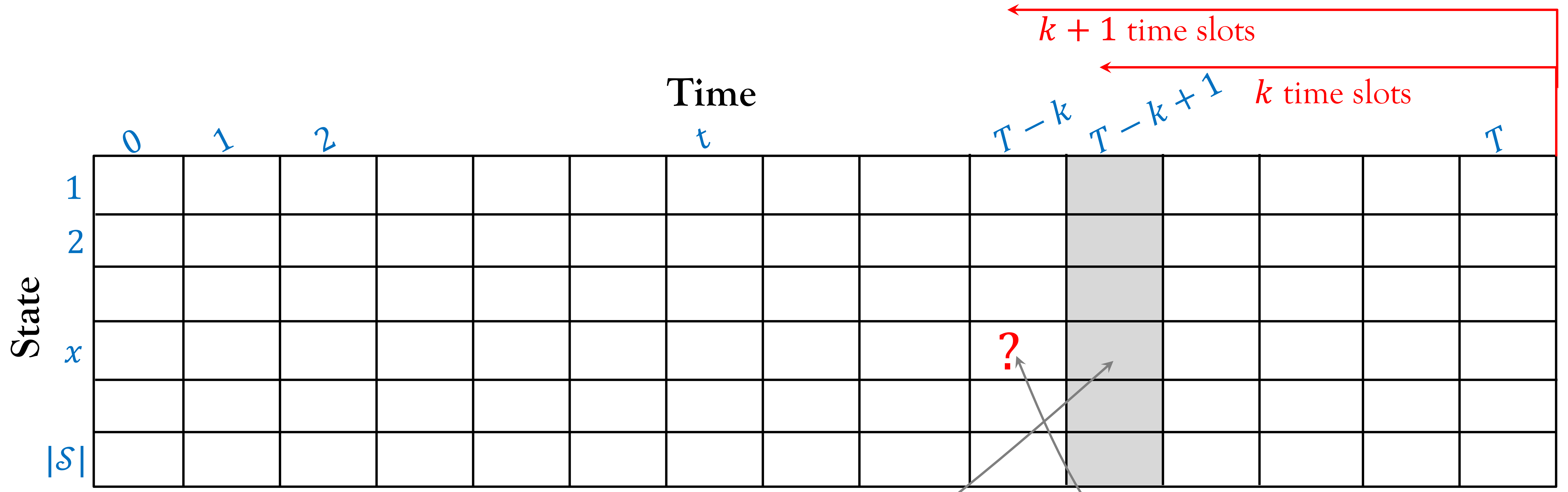
- Each column of the table is associated with a time t.
- Each row of the table is associated with a state x. In the table, we are representing a state with a scalar. This does not mean that the state can't be a tuple. You can simply consider these scalars as the **enumerate version of a state** (refer lecture 10 slides).

Policy Evaluation: IPE and Bellman Equation



- Each column of the table is associated with a time t. t ranges from 0 to T.
- Each row of the table is associated with a state x.
- The cell corresponding to column t (0 indexed) and state x (1 indexed) will contain the value of $V^{\pi}_{T-t+1,T}(x)$ where $T-t+1$ is the number of time slots backwards from horizon T.

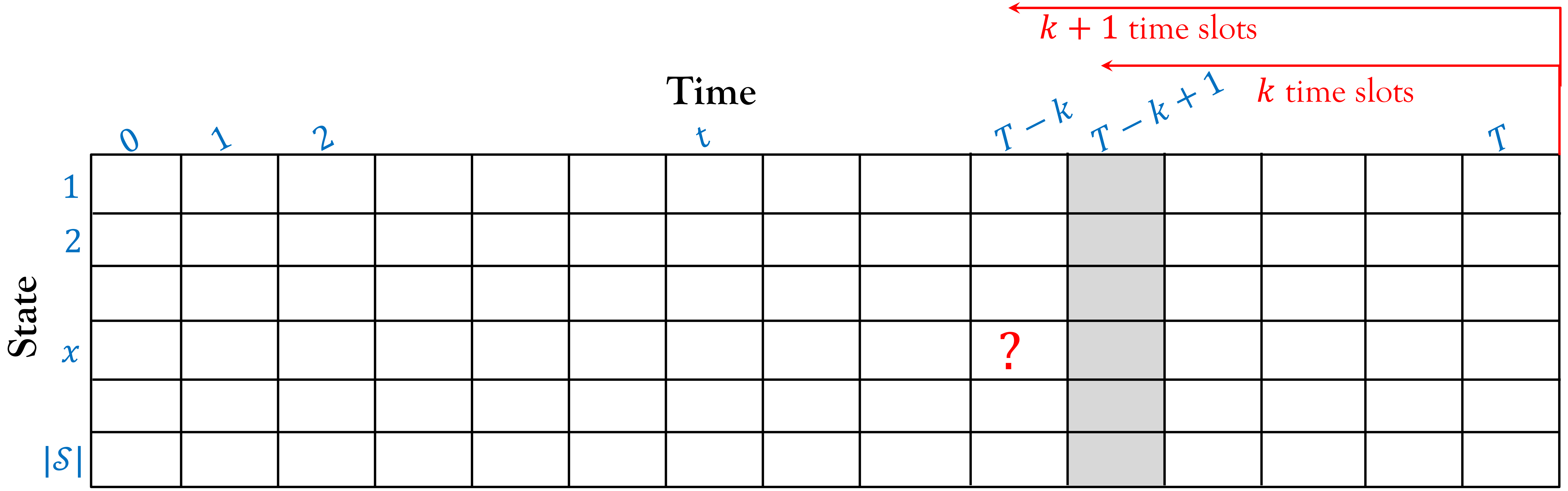
Policy Evaluation: IPE and Bellman Equation



- In many of the DP problems, we have to separately inspect the edge case and the non-edge case. We deal with the non-edge cases first.

- Suppose we are given $V^{\pi}_{k,T}(x)$ for all x. We now show that for this DP problem, we can compute $V^{\pi}_{k,+1\ T}(x)$ if we are given $\boldsymbol{V^{\pi}_{k,T}(x')}$ **for all** $\boldsymbol{x'}$.

Policy Evaluation: IPE and Bellman Equation

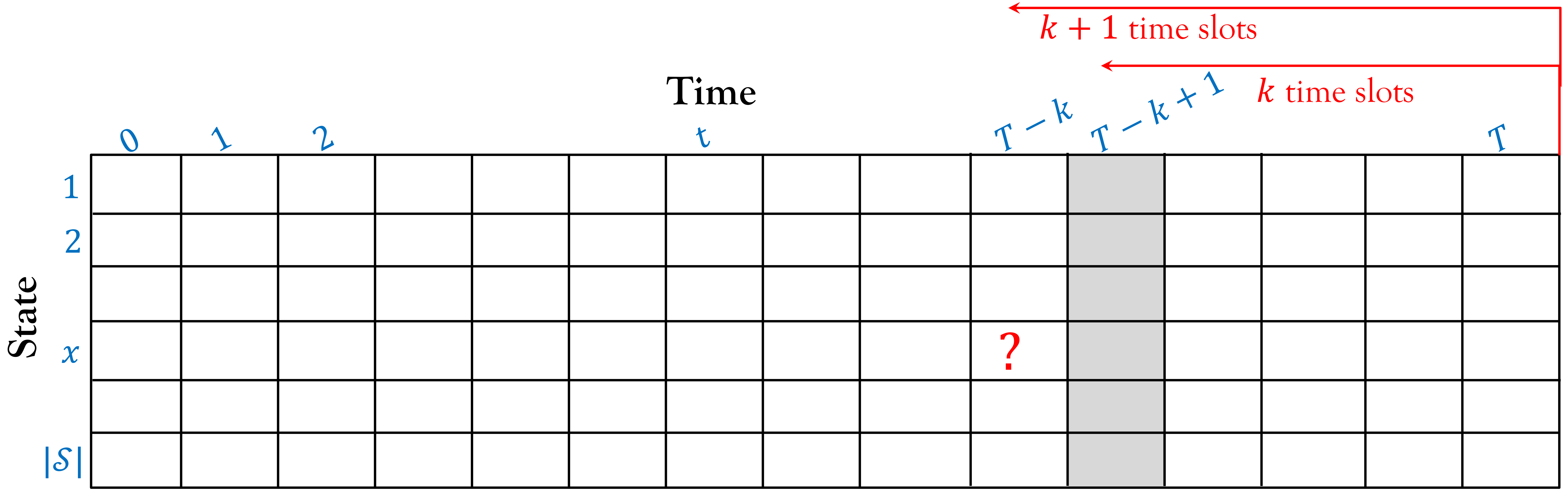


➢ Suppose the state is x at time $T-k$ ($k+1$ times slots back from the last time slot) and **agent takes action $\boldsymbol{a}$**. Then the agents expected return is,

$$r(x,a)+\beta\sum_{x'\in\mathcal{S}}P[x'\mid x,a]\,V^{\pi}_{k,T}(x')$$

We now explain the above expression.

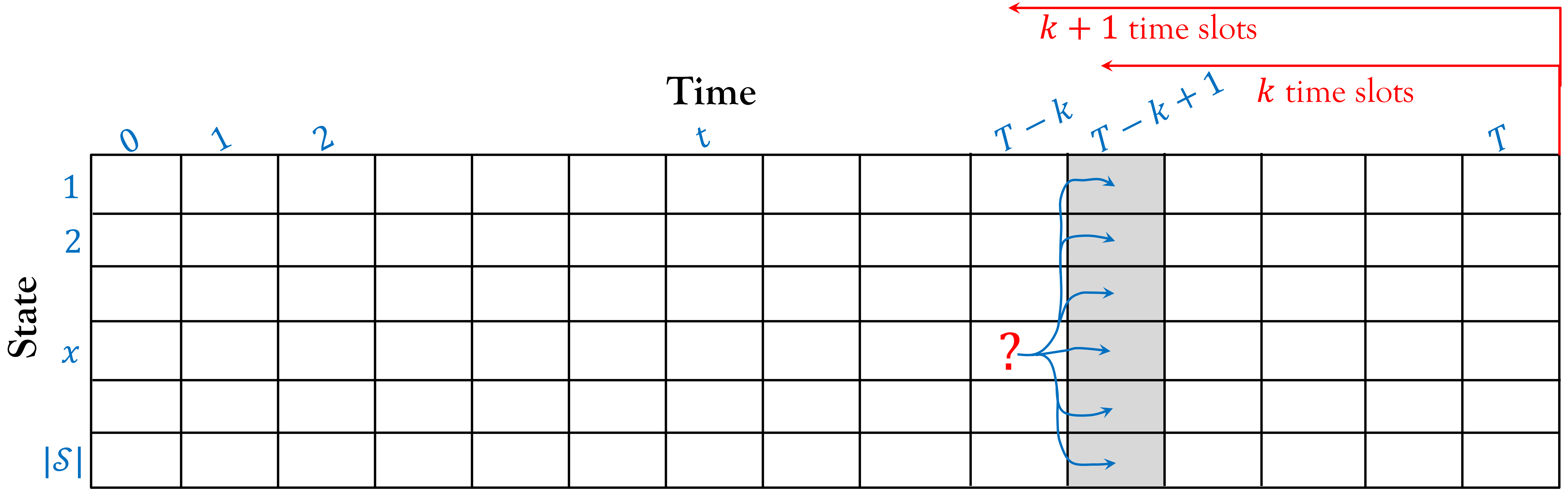
Policy Evaluation: IPE and Bellman Equation



- If the state is x at time $T-k$ and agent takes action a:
  - Agent earns an "immediate reward" r at time $T-k$ whose average is $r(x,a) = \sum_{\tilde{r}} P[\tilde{r} \mid x,a]$ where $P[\tilde{r} \mid x,a]$ is the reward probability distribution.

$$r(x,a) + \beta \sum_{x' \in \mathcal{S}} P[x' \mid x,a]\, V_{k,T}^{\pi}(x')$$

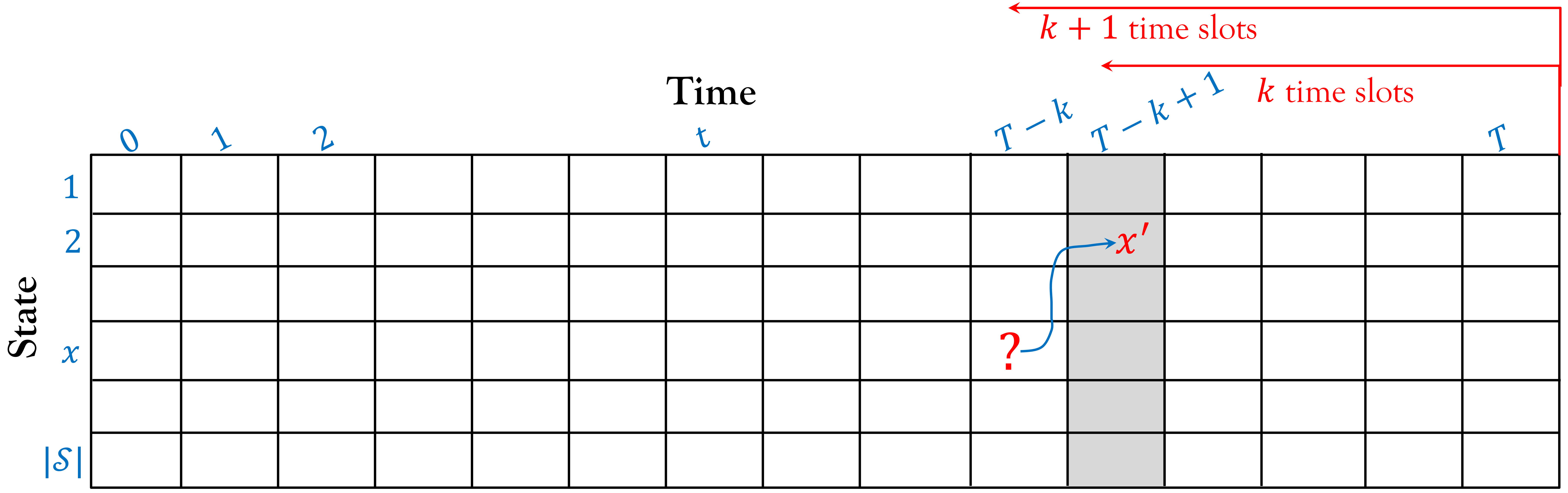
Policy Evaluation: IPE and Bellman Equation



- If the state is x at time $T-k$ and agent takes action a:
  - The environment transitions to state x' in time slot $T-k+1$ where the probability of transitioning to state x' is $P[x' \mid x, a]$ (state transition probability distribution).

$$r(x,a) + \beta \sum_{x' \in \mathcal{S}} P[x' \mid x, a]\, V^{\pi}_{k,T}(x')$$

Policy Evaluation: IPE and Bellman Equation



- If the state is x at time $T-k$ and agent takes action a:
  - Consider a particular state x'. The expected value of the return starting from x' at time $T-k+1$ is $V^{\pi}_{k,T}(x')$ (which is given to us; remember $V^{\pi}_{k,T}(x)$ for all x is given). But, why this additional β?

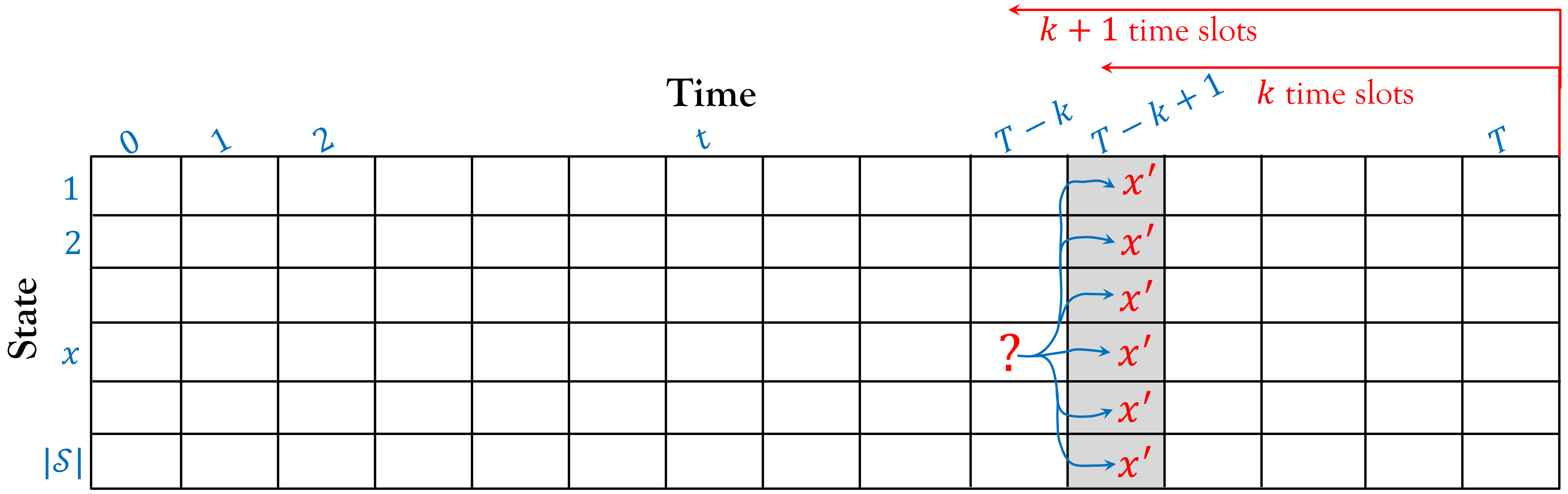
$$r(x,a) + \boxed{\beta} \sum_{x' \in \mathcal{S}} P[x' \mid x, a] \boxed{V^{\pi}_{k,T}(x')}$$

Policy Evaluation: IPE and Bellman Equation

$$\begin{aligned} G_{k+1,T} &= r_{T-k} + \beta r_{T-k+1} + \beta^2 r_{T-k+2} + \cdots + \beta^k r_T \\ &= r_{T-k} + \beta\left(r_{T-k+1} + \beta r_{T-k+2} + \cdots + \beta^{k-1} r_T\right) \\ &= r_{T-k} + \beta G_{k,T} \end{aligned}$$

- $V^{\pi}_{k+1,T}(x)$ which we want to compute is the expected value of $G_{k+1,T}$.
- $r(x,a)$ which we discussed three slides back is the expected value of r_{T-k}.
- $V^{\pi}_{k,T}(x')$ which is given is the expected value of $G_{k,T}$.
- And here is β. By comparing the overall structure, you can get an intuition. And yes, this is only an intuition!

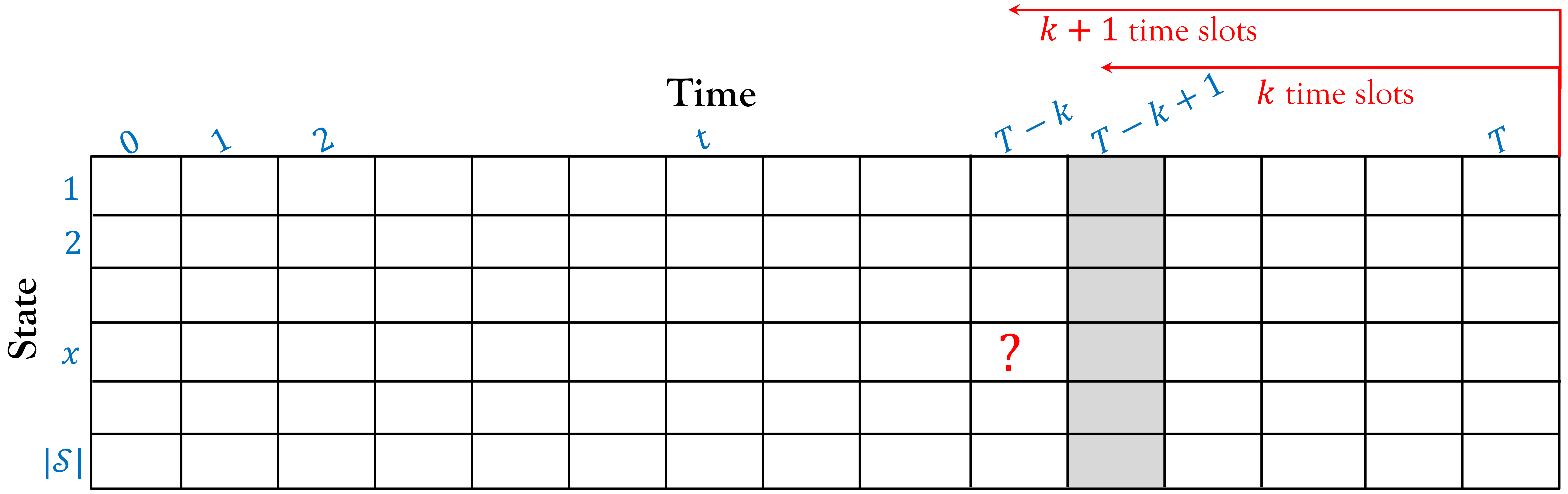
Policy Evaluation: IPE and Bellman Equation



- If the state is x at time $T-k$ and agent takes action a:
  - To get the "future reward", we have to sum over all the next states x' in the state space the environment can transition to.

$$r(x,a) + \beta \sum_{x' \in \mathcal{S}} P[x' \mid x, a]\, V^{\pi}_{k,T}(x')$$

Policy Evaluation: IPE and Bellman Equation



- If the state is x at time $T-k$ and agent takes action a:
  - The expression below is of $q^{\pi}_{k+1,T}(x,a)$ and not $V^{\pi}_{k+1,T}(x)$ because action a at time $T-k$ is fixed. Now, we will derive an expression for $V^{\pi}_{k+1,T}(x)$ in terms of $q^{\pi}_{k+1,T}(x,a)$.

$$q^{\pi}_{k+1,T}(x,a) = r(x,a) + \beta \sum_{x' \in \mathcal{S}} P[x' \mid x,a]\, V^{\pi}_{k,T}(x')$$

Policy Evaluation: IPE and Bellman Equation

- To derive an expression for $V_{k+1,T}^{\pi}(x)$ in terms of $q_{k+1,T}^{\pi}(x,a)$ we just have to choose action a using policy π. This is because for value function, even the action in the first time slot is taken according to policy π.

- The probability of choose action a in time slot $T-k$ when the state is x is $\pi(a \mid x)$.

- Hence, in order to compute $V_{k+1,T}^{\pi}(x)$ we have to sum $q_{k+1,T}^{\pi}(x,a)$ over all action a in the action space $\mathcal{A}(x)$ weighted by the probability of taking action a which is $\pi(a \mid x)$. Hence,

$$V_{k+1,T}^{\pi}(x) = \sum_{a \in \mathcal{A}(x)} \pi(a \mid x)\, q_{k+1,T}^{\pi}(x,a)$$

Policy Evaluation: IPE and Bellman Equation

- So we have,

$$V^{\pi}_{k+1,T}(x) = \sum_{a \in \mathcal{A}(x)} \pi(a \mid x)\, q^{\pi}_{k+1,T}(x,a) \qquad \text{(E.2.)}$$

- We have also derived,

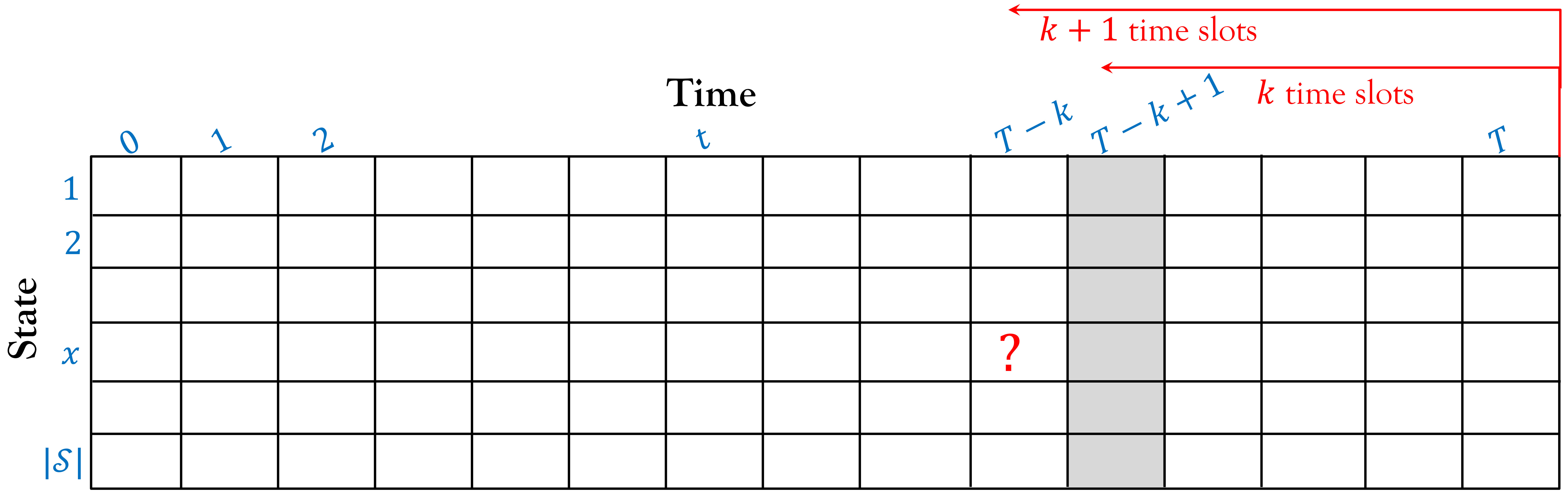
$$q^{\pi}_{k+1,T}(x,a) = r(x,a) + \beta \sum_{x' \in \mathcal{S}} P[x' \mid x, a]\, V^{\pi}_{k,T}(x') \qquad \text{(E.3.)}$$

- Substituting $q^{\pi}_{k+1,T}(x,a)$ from (E.3.) to (E.2) we get,

$$V^{\pi}_{k+1,T}(x) = \sum_{a \in \mathcal{A}(x)} \pi(a \mid x) \left(r(x,a) + \beta \sum_{x' \in \mathcal{S}} P[x' \mid x, a]\, V^{\pi}_{k,T}(x') \right) \qquad \text{(E.4.)}$$

NOTE: Derivation of equations (E.2.) to (E.4.) that we did here is not rigorous. The rigorous version of the derivation is in the appendix of these slides (click here).

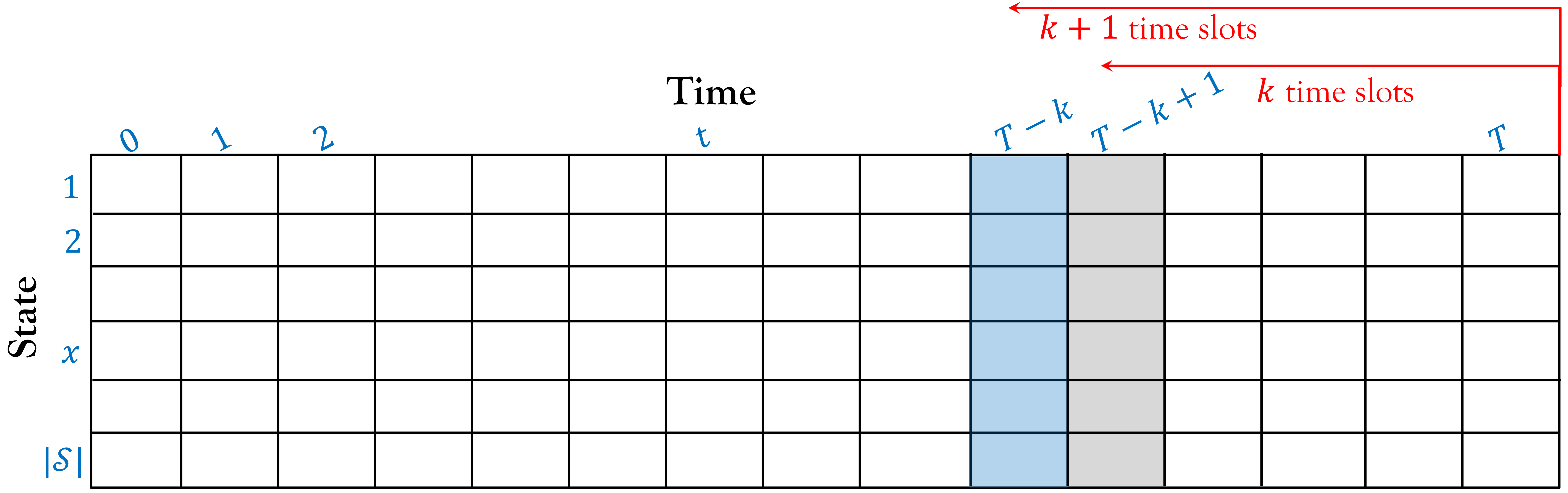
Policy Evaluation: IPE and Bellman Equation



$$V^{\pi}_{k+1,T}(x) = \sum_{a \in \mathcal{A}(x)} \pi(a \mid x) \left(r(x,a) + \beta \sum_{x' \in \mathcal{S}} P[x' \mid x, a]\, V^{\pi}_{k,T}(x') \right)$$

- The above is the formula to compute $V^{\pi}_{k+1,T}(x)$ given $V^{\pi}_{k,T}(x')$ for all x'. So, essentially we can compute **an element of the $(k+1)^{th}$ column** from the last time slot using k^{th} column from the last time slot.

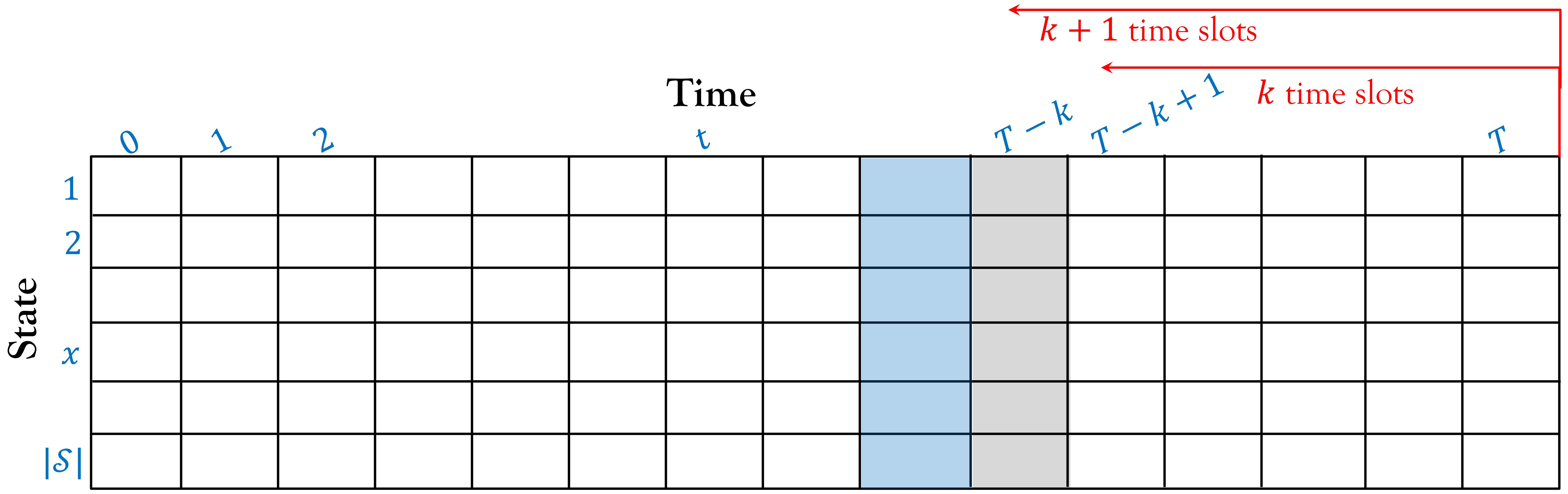
Policy Evaluation: IPE and Bellman Equation



$$V^{\pi}_{k+1,T}(x) = \sum_{a\in\mathcal{A}(x)} \pi(a\,|\,x)\left(r(x,a) + \beta \sum_{x'\in\mathcal{S}} P[x'\,|\,x,a]\, V^{\pi}_{k,T}(x')\right)$$

- The above is the formula to compute $V^{\pi}_{k+1,T}(x)$ given $V^{\pi}_{k,T}(x')$ for all x'. So, essentially we can compute **an element of the $(k+1)^{th}$ column** from the last time slot using k^{th} column from the last time slot. We can use this same formula to compute the **entire $(k+1)^{th}$ column**.

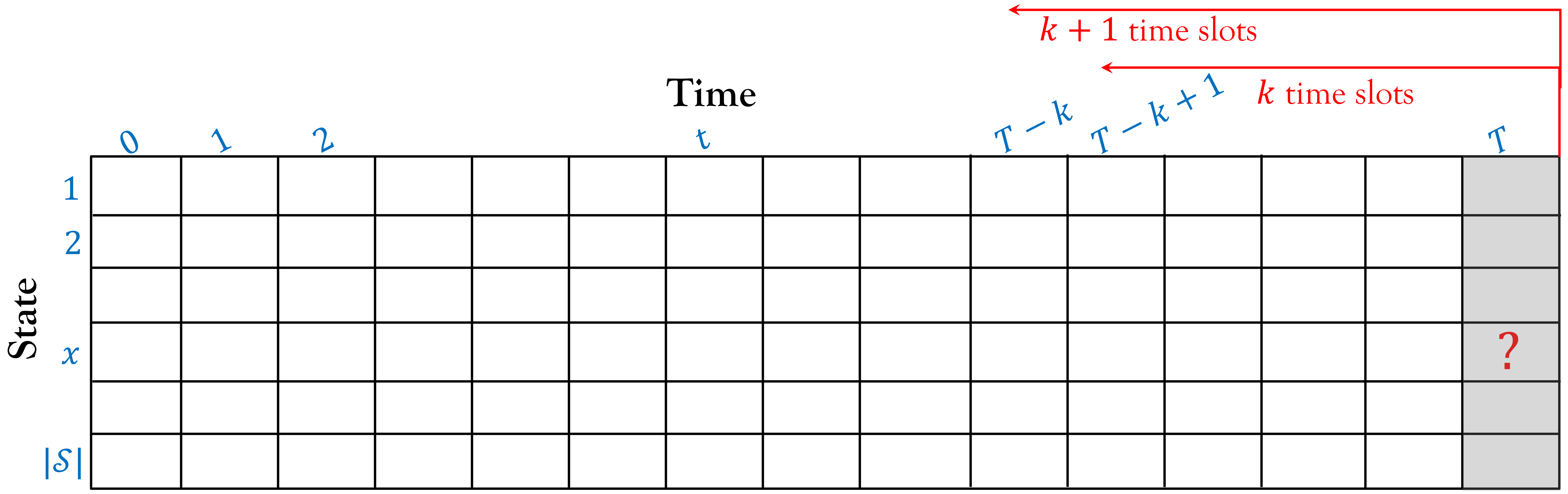
Policy Evaluation: IPE and Bellman Equation



$$V^{\pi}_{k+1,T}(x) = \sum_{a \in \mathcal{A}(x)} \pi(a \mid x) \left(r(x,a) + \beta \sum_{x' \in \mathcal{S}} P[x' \mid x, a] \, V^{\pi}_{k,T}(x') \right)$$

- Similarly after computing the entire $(k+1)^{th}$ column we can use it to compute the entire $(k+2)^{th}$ column. And so on… Finally, all the **non-edge** cases done.

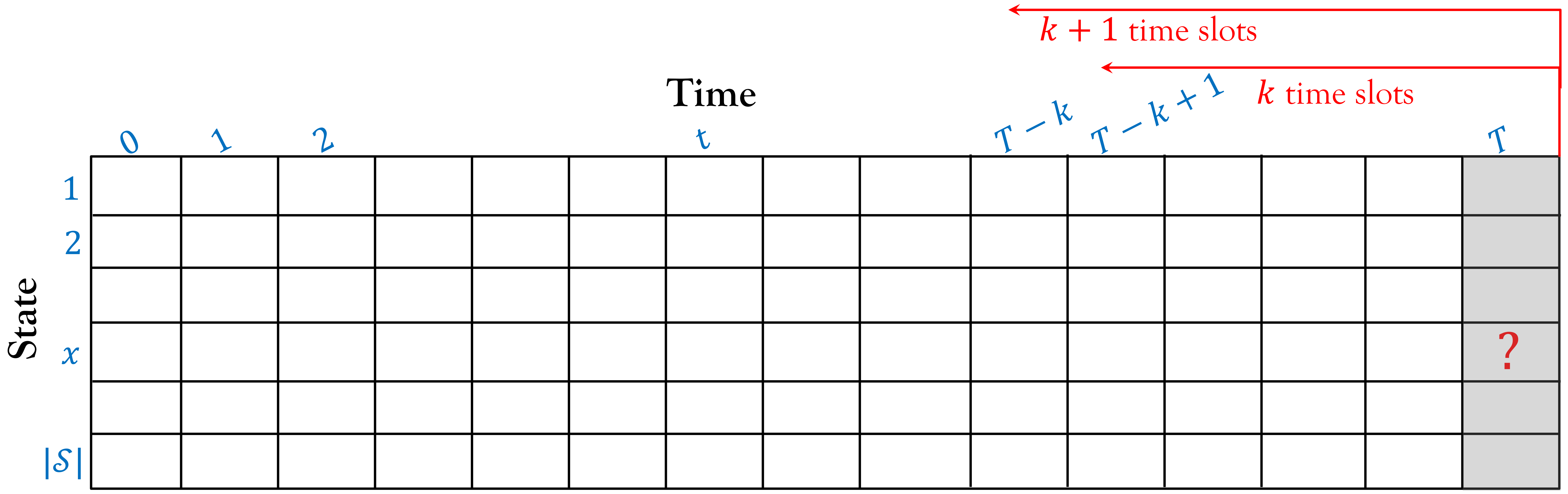
Policy Evaluation: IPE and Bellman Equation



$$V^{\pi}_{k+1,T}(x) = \sum_{a \in \mathcal{A}(x)} \pi(a \mid x) \left(r(x,a) + \beta \sum_{x' \in \mathcal{S}} P[x' \mid x,a]\, V^{\pi}_{k,T}(x') \right)$$

- The **edge case** for this DP problem is the **last column**, i.e. $V^{\pi}_{1,T}(x)$ for all x. It is an edge case because $V^{\pi}_{0,T}(x')$ is not defined an hence we can't use the above formula.

Policy Evaluation: IPE and Bellman Equation



$$V^{\pi}_{k+1,T}(x) = \sum_{a \in \mathcal{A}(x)} \pi(a \mid x) \left(r(x,a) + \beta \sum_{x' \in \mathcal{S}} P[x' \mid x, a]\, V^{\pi}_{k,T}(x') \right)$$

- For the last column, there is **no future reward** because the optimization horizon ends after time T. Hence, we can simply ignore this term. This is also equivalent to setting $V^{\pi}_{0,T}(x') = 0$ for all x'. So, we now have the recipe to compute the entire table.

Policy Evaluation: IPE and Bellman Equation

$$V^{\pi}_{k+1,T}(x) = \sum_{a \in \mathcal{A}(x)} \pi(a \mid x)\, q^{\pi}_{k+1,T}(x,a) \tag{E.2.}$$

$$q^{\pi}_{k+1,T}(x,a) = r(x,a) + \beta \sum_{x' \in \mathcal{S}} P[x' \mid x,a]\, V^{\pi}_{k,T}(x') \tag{E.3.}$$

$$V^{\pi}_{k+1,T}(x) = \sum_{a \in \mathcal{A}(x)} \pi(a \mid x) \left(r(x,a) + \beta \sum_{x' \in \mathcal{S}} P[x' \mid x,a]\, V^{\pi}_{k,T}(x') \right) \tag{E.4.}$$

- The above formulas are meant for finite optimization horizon. But, what about infinite optimization horizon which is our main goal? What happens to the above formulas when $T \to \infty$?

Policy Evaluation: IPE and Bellman Equation

- As $T \to \infty$, $V^{\pi}_{k,T}(x) \to V^{\pi}(x)$ because of the stationarity and infinite series reason mentioned in this slide. $V^{\pi}_{k,T}(x) \to V^{\pi}(x)$ implies that the value function is not dependent on k.

- Similarly, as $T \to \infty$, $q^{\pi}_{k,T}(x,a) \to q^{\pi}(x,a)$. Based on these observations we get,

$$V^{\pi}(x) = \sum_{a \in \mathcal{A}(x)} \pi(a \mid x) q^{\pi}(x,a) \tag{E.5.}$$

$$q^{\pi}(x,a) = r(x,a) + \beta \sum_{x' \in \mathcal{S}} P[x' \mid x,a] V^{\pi}(x) \tag{E.6.}$$

Bellman equation

$$\boxed{V^{\pi}(x) = \sum_{a \in \mathcal{A}(x)} \pi(a \mid x)\left(r(x,a) + \beta \sum_{x' \in \mathcal{S}} P[x' \mid x,a] V^{\pi}(x') \right)} \tag{E.7.}$$

Policy Evaluation: IPE and Bellman Equation

- Equation (E.7.) is the famous Bellman equation (not to confuse with Bellman optimality equation which we will encounter later in these slides).

- Bellman equation gives us an expression for the value function $V^{\pi}(x)$. But now we need a way to solve (E.7.) and hence compute $V^{\pi}(x)$. So, how to solve (E.7.)?

Policy Evaluation: IPE and Bellman Equation

- Equation (E.7.) is the famous Bellman equation (not to confuse with Bellman optimality equation which we will encounter later in these slides).

- Bellman equation gives us an expression for the value function $V^{\pi}(x)$. But now we need a way to solve (E.7.) and hence compute $V^{\pi}(x)$. So, how to solve (E.7.)?

  - We use equation (E.4.). Consider $T = \infty$ in (E.4.) and keep iterating using the following formula (which is SAME as (E.4.); just removed T and π from V),

$$V_{k+1}(x) = \sum_{a \in \mathcal{A}(x)} \pi(a \mid x) \left(r(x,a) + \beta \sum_{x' \in \mathcal{S}} P[x' \mid x, a] \, V_k(x') \right) \qquad \text{(E.8.)}$$

    Equation (E.8.) is called iterative policy evaluation (IPE).

  - The iteration stops when the convergence of $V_k(x)$ is detected for all x. How to detect convergence? There is no one answer. One possible strategy is to iterate until $|V_{k+1}(x) - V_k(x)|$ is below a specified threshold for all x.

Policy Evaluation: Psuedocode for IPE

```
Given: A policy, $\pi$, and a threshold, $\theta$.

(S1): For all $x \in \mathcal{S}$, initialize $V(x)$ arbitrarily to any real value. Initialize $\Delta = \infty$.

(S2): while $\Delta > \theta$:

(S3):         for all $x \in \mathcal{S}$:

(S4):                 $V_{new}(x) = \sum_{a \in \mathcal{A}(x)} \pi(a \mid x) \cdot \left( r(x, a) + \beta \sum_{x' \in \mathcal{S}} P[x' \mid x, a] \, V(x') \right)$

(S5):         Compute $\Delta = \max_{x \in \mathcal{S}} |V_{k+1}(x) - V_k(x)|$.

(S6):         Set $V(x) = V_{new}(x)$ for all $x \in \mathcal{S}$.

(S7): Return $V(x)$ for all $x \in \mathcal{S}$.
```

Computing Optimal Policy

- A policy π^* is optimal if and only if,

$$V^{\pi^*}(x) \geq V^{\pi}(x) \, , \forall x \in \mathcal{S} \qquad \text{(E.9.)}$$

The above equation essentially means that the value function corresponding to the optimal policy is greater than or equal to value function corresponding to any other policy **for all the states**.

- There can be more than one policies that are optimal. We want to find one of those optimal policies.

Theorem 1: If an MDP has a finite discrete state space and finite discrete action space (for all the states), then **there exists a deterministic policy** that is optimal.

Proof: There is [2]. It is OPTIONAL to check it.

[2] Martin L. Putterman, "Markov Decision Processes: Discrete Stochastic Dynamic Programming", Chapter 6 (Discounted Markov Decision Problems).

Computing Optimal Policy

- The importance of Theorem 1 is that it **reduces the search for optimal policy to only deterministic policies**.
- A deterministic policy can be written as $\pi(x)$, i.e. the (deterministic) action corresponding to state x.
- How can we use Theorem 1 to find the optimal policy?

Computing Optimal Policy

- The importance of Theorem 1 is that it **reduces the search for optimal policy to only deterministic policies**.
- A deterministic policy can be written as $\pi(x)$, i.e. the (deterministic) action corresponding to state x.
- How can we use Theorem 1 to find the optimal policy?

Answer: Iterate over all the deterministic policies. For each of these deterministic policy, π, use iterative policy evaluation to compute $V^{\pi}(x)$ for all $x \in \mathcal{S}$. Find the policy that satisfies (E.9.).

The problem of this approach is that the **number of deterministic policy is exponential in the number of states**. The number of deterministic policies are the total number of permutations of the actions available in each state which is,

$$|\mathcal{A}(x_1)| \times |\mathcal{A}(x_2)| \times \cdots \times \left|\mathcal{A}\left(x_{|\mathcal{S}|}\right)\right|$$

Note: $\times$ in the above formula is simple multiplication.

Computing Optimal Policy

- In order to come up with better approaches to compute the optimal policy we need to define a few concepts first.

Definition (Optimal Value function): The optimal value function of state, x, is the maximum value of the expected return G_t, given that $x_t = x$ (i.e. starting from state x at time t). We denote it using $V^*(x)$.

Definition (Optimal Action-Value function): The optimal action-value function of state action pair, (x, a), is the maximum value of the expected return G_t, given that $x_t = x$ (i.e. starting from state x at time t) and $a_t = a$ (i.e. taking action a at start time t; action a may not be optimal for state x). We denote it using $q^*(x, a)$.

Computing Optimal Policy

- In order to come up with better approaches to compute the optimal policy we need to define a few concepts first.

Definition (Optimal Value function for finite optimization horizon): The optimal value function for finite optimization horizon corresponding to state, x, is the maximum value of the expected return $G_{k,T}$, given that $x_{T-k+1} = x$ (i.e. starting from state x at time $T-k+1$). We denote it using $V^*_{k,T}(x)$.

Definition (Optimal Action-Value function for finite optimization horizon): The optimal action-value function for finite optimization horizon corresponding to state action pair, (x, a), is the maximum value of the expected return $G_{k,T}$, given that $x_{T-k+1} = x$ (i.e. starting from state x at time $T-k+1$) and $a_{T-k+1} = a$ (i.e. taking action a at start time $T-k+1$; action a may not be optimal for state x). We denote it using $q^*_{k,T}(x, a)$.

Computing Optimal Policy

- By definition of optimal value function and optimal action-value function, $q^*(x, a)$, $V^*(x)$ and $q^*(x, a)$ are related as follows,

$$V^*(x) = \max_{a \in \mathcal{A}(x)} q^*(x, a) \tag{E.10.}$$

Consequently, the optimal policy is,

$$\pi^*(x) = \operatorname*{argmax}_{a \in \mathcal{A}(x)} q^*(x, a) \tag{E.11.}$$

Computing Optimal Policy

- By definition of optimal value function and optimal action-value function, $q^*(x,a)$, $V^*(x)$ and $q^*(x,a)$ are related as follows,

$$V^*(x) = \max_{a\in\mathcal{A}(x)} q^*(x,a) \qquad \text{(E.10.)}$$

  Consequently, the optimal policy is,

$$\pi^*(x) = \operatorname*{argmax}_{a\in\mathcal{A}(x)} q^*(x,a) \qquad \text{(E.11.)}$$

- Similarly, by definition of optimal value function and optimal action-value function **for finite horizon**, $V^*_{k,T}(x)$ and $q^*_{k,T}(x,a)$ are related as follows,

$$V^*_{k,T}(x) = \max_{a\in\mathcal{A}(x)} q^*_{k,T}(x,a) \qquad \text{(E.12.)}$$

  Consequently, the optimal policy is,

$$\pi^*_{k,T}(x) = \operatorname*{argmax}_{a\in\mathcal{A}(x)} q^*_{k,T}(x,a) \qquad \text{(E.13.)}$$

Computing Optimal Policy

- By definition of optimal value function and optimal action-value function, $q^*(x,a)$, $V^*(x)$ and $q^*(x,a)$ are related as follows,

$$V^*(x) = \max_{a \in \mathcal{A}(x)} q^*(x,a) \tag{E.10.}$$

Consequently, the optimal policy is,

$$\pi^*(x) = \underset{a \in \mathcal{A}(x)}{\operatorname{argmax}}\ q^*(x,a) \tag{E.11.}$$

Critical observation: Equation (E.11.) suggest that in order to compute the optimal policy, we just have to compute $q^*(x,a)$. Then, we use (E.11.) to compute the optimal policy.

Computing Optimal Policy

- Now $q^*(x, a)$ can be can be computed using $V^*(x)$ using an equation very similar to equation (E.6.) of this slide. So, we concentrate on computing $V^*(x)$.

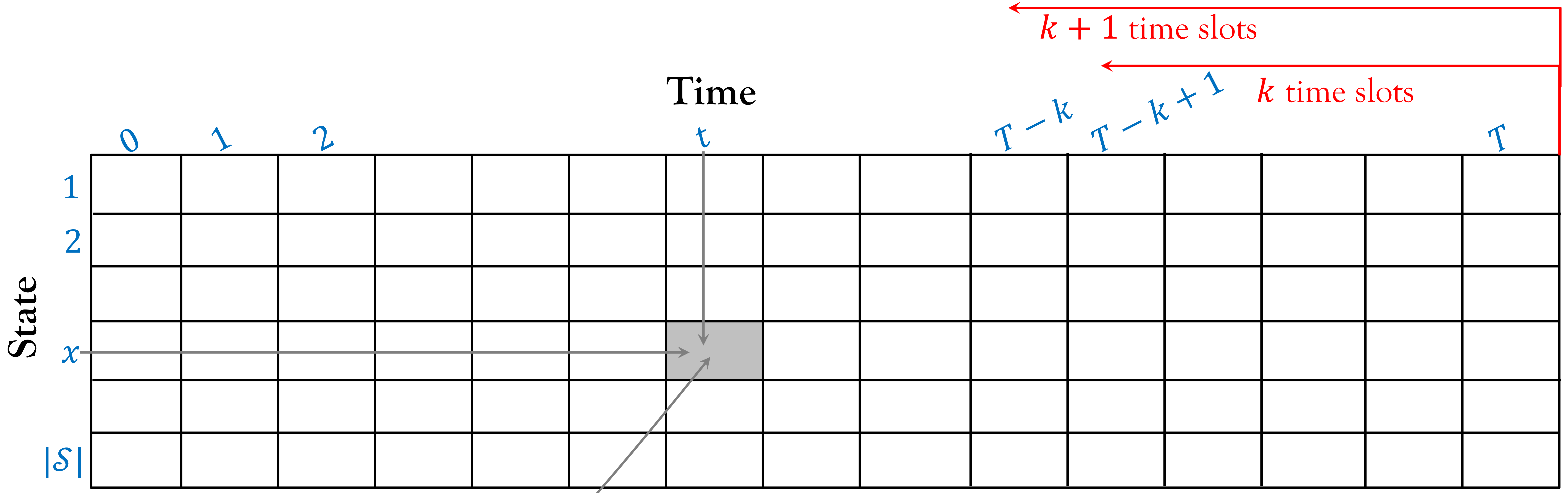
- The approach to compute $V^*(x)$ is very similar to compute $V^\pi(x)$ while doing policy evaluation. The steps are as follows:

  - First, we derive an expression to compute the optimal value function, $V_{k,T}^*(x)$, for the finite horizon case. This expression that we derive is called **value iteration (VI)**.

  - Second, we then use the expression for $V_{k,T}^*(x)$ to derive an expression for $V^*(x)(x)$. This expression is called the **Bellman optimality equation**.

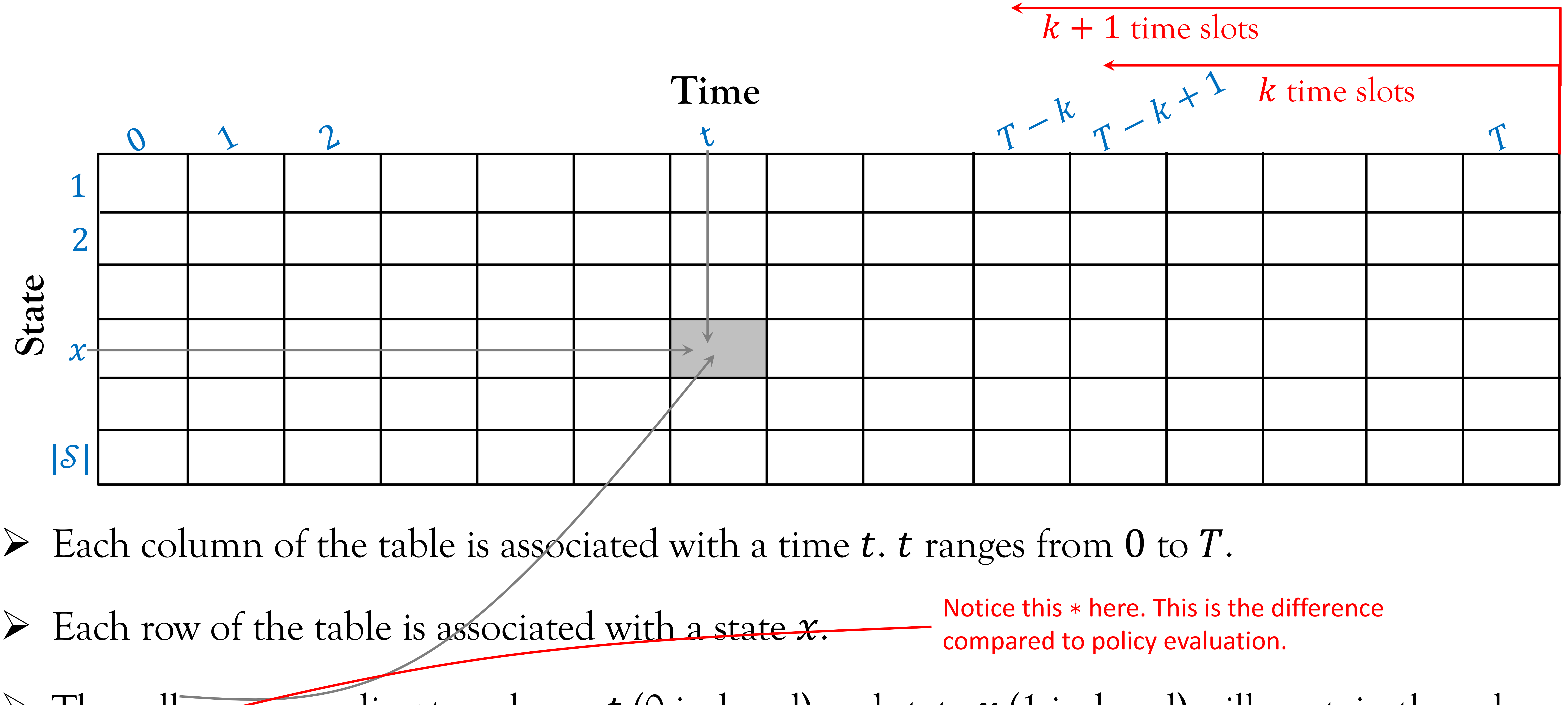
NOTE: Since the approach is very similar to policy evaluation, I will not list out individual steps in great details.

Computing Optimal Policy: Bellman Optimality Equation and VI



- Each column of the table is associated with a time t. t ranges from 0 to T.
- Each row of the table is associated with a state x.
- The cell corresponding to column t (0 indexed) and state x (1 indexed) will contain the value of $V^*_{T-t+1,T}(x)$ where $T-t+1$ is the number of time slots backwards from horizon T.

Computing Optimal Policy: Bellman Optimality Equation and VI

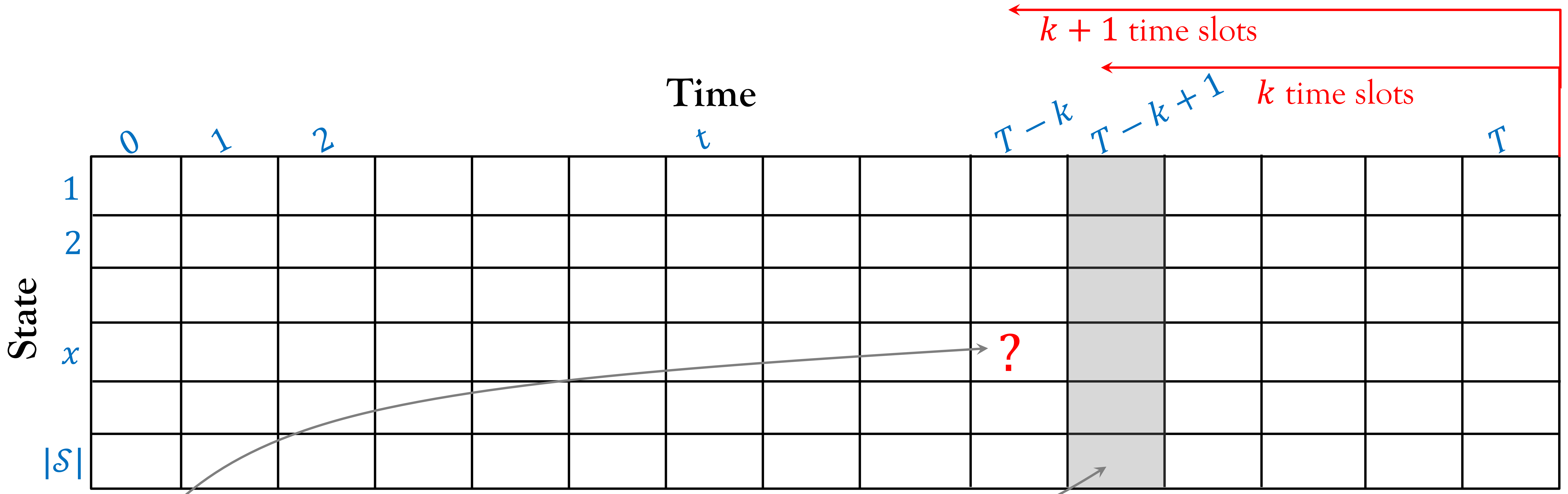


- Each column of the table is associated with a time t. t ranges from 0 to T.
- Each row of the table is associated with a state x.

Notice this $*$ here. This is the difference compared to policy evaluation.

- The cell corresponding to column t (0 indexed) and state x (1 indexed) will contain the value of $V^*_{T-t+1,T}(x)$ where $T-t+1$ is the number of time slots backwards from horizon T.

Computing Optimal Policy: Bellman Optimality Equation and VI



- Suppose we are given $V^*_{k,T}(x)$ for all x. We show that for this DP problem, we can compute $V^*_{k+1,T}(x)$ if we are given $V^*_{k,T}(x')$ for all x'.
- In order to compute $V^*_{k+1,T}(x)$, it is enough to compute $q^*_{k+1,T}(x,a)$. Then, $V^*_{k+1,T}(x)$ can be computed using equation (E.12.) of this slide.

Computing Optimal Policy: Bellman Optimality Equation and VI

$$q^*_{k+1,T}(x,a) = r(x,a) + \beta \sum_{x' \in \mathcal{S}} P[x' \mid x, a] \, \boxed{V^*_{k,T}(x')}$$

The expression for $q^*_{k+1,\,T}(x,a)$ is given below. We now explain this expression.

- Everything is same as policy evaluation except this term which is $V^*_{k,T}(x')$ instead of $V^{\pi}_{k,T}(x')$.

- To understand why, let's recall the definition of $q^*_{k+1,T}(x,a)$:

$q^*_{k+1,T}(x,a)$ is the **maximum value of the expected return $\boldsymbol{G_{k+1,T}}$**, given that $\boldsymbol{x_{T-k} = x}$ **and** $\boldsymbol{a_{T-k} = a}$.

- Given that the state and action at time $T-k$ is fixed at x and a respectively, **maximizing the expected return $\boldsymbol{G_{k+1,T}}$** is **equivalent** to **maximizing the expected return from next time slot $\boldsymbol{T-k+1}$ onwards**.

- Now if the state at time $T-k+1$ is x', the maximum value of the expected return from time slot $T-k+1$ is $V^*_{k,T}(x')$. This completed the explation.

Computing Optimal Policy: Bellman Optimality Equation and VI

- So from this slide we have,

$$V^*_{k+1,T}(x) = \max_{a \in \mathcal{A}(x)} q^*_{k+1,T}(x,a) \tag{E.12.}$$

- We just now derived,

$$q^*_{k+1,T}(x,a) = r(x,a) + \beta \sum_{x' \in \mathcal{S}} P[x' \mid x,a]\, V^*_{k,T}(x') \tag{E.14.}$$

- Substituting $q^*_{k+1,T}(x,a)$ from (E.14.) to (E.12) we get,

$$V^*_{k+1,T}(x) = \max_{a \in \mathcal{A}(x)} \left(r(x,a) + \beta \sum_{x' \in \mathcal{S}} P[x' \mid x,a]\, V^*_{k,T}(x') \right) \tag{E.15.}$$

NOTE: Derivation of equation (E.14.) that we did here is not rigorous. The rigorous version of the derivation is in the appendix of these slides (click here).

Computing Optimal Policy: Bellman Optimality Equation and VI

- The formulas in the previous slides are meant for finite optimization horizon. In order to extend them to infinite optimization horizon, we have to let $T \to \infty$.

- As $T \to \infty$, $V^*_{k,T}(x) \to V^*(x)$ and $q^*_{k,T}(x,a) \to q^*(x,a)$ for all k's. The reason is same as that in policy evaluation (stationary distribution and infinite series). Based on these observations, equations (E.12.), (E.14.), and (E.15.) can be modified as follows for infinite horizon.

$$V^*(x) = \max_{a \in \mathcal{A}(x)} q^*(x,a) \qquad \text{(E.16.)}$$

$$q^*(x,a) = r(x,a) + \beta \sum_{x' \in \mathcal{S}} P[x' \mid x,a]\, V^*(x') \qquad \text{(E.17.)}$$

Bellman optimality equation

$$\boxed{V^*(x) = \max_{a \in \mathcal{A}(x)} \left(r(x,a) + \beta \sum_{x' \in \mathcal{S}} P[x' \mid x,a]\, V^*(x') \right)} \qquad \text{(E.18.)}$$

Computing Optimal Policy: Bellman Optimality Equation and VI

- Equation (E.18.) is the famous Bellman optimality equation.

- Now, let's pause for a moment and recall what we did till now and what to do next.
  - We want to compute the optimal policy $\pi^*(x)$. In this slide (click here) we discussed that if we know $q^*(x, a)$ we can compute $\pi^*(x)$ using equation (E.11.).
  - Then in this slide (click here) we indicated that $q^*(x, a)$ can be computed using $V^*(x)$. Equation (E.17.) can be used to compute $q^*(x, a)$ using $V^*(x)$.
  - So essentially, all that is left is to compute $V^*(x)$. If we can compute $V^*(x)$, we can find the optimal policy $\pi^*(x)$.

Computing Optimal Policy: Bellman Optimality Equation and VI

- Bellman optimality equation (equation (E.18.)) gives us an expression for the optimal value function $V^*(x)$. But now we need a way to solve (E.18.) and hence compute $V^*(x)$. So, **how to solve (E.18.)?**

Computing Optimal Policy: Bellman Optimality Equation and VI

- Bellman optimality equation (equation (E.18.)) gives us an expression for the optimal value function $V^*(x)$. But now we need a way to solve (E.18.) and hence compute $V^*(x)$. So, **how to solve (E.18.)?** Answer: Similar to iterative policy evaluation.

  - We use equation (E.15.). Consider $T = \infty$ in (E.15.) and keep iterating using the following formula (which is SAME as (E.15.); just removed T and $*$ from V),

$$V_{k+1}(x) = \max_{a \in \mathcal{A}(x)} \left(r(x,a) + \beta \sum_{x' \in \mathcal{S}} P[x' \mid x, a] V_k(x') \right) \qquad \text{(E.19.)}$$

    Equation (E.19.) is called **value iteration (VI)**.

  - The iteration stops when the convergence of $V_k(x)$ is detected for all x. How to detect convergence? There is no one answer. One possible strategy is to iterate until $|V_{k+1}(x) - V_k(x)|$ is below a specified threshold for all x.

Computing Optimal Policy : Psuedocode for VI

Given: A threshold, θ.

```
(S1): For all $x \in \mathcal{S}$, initialize $V(x)$ arbitrarily to any real value. Initialize $\Delta = \infty$.
(S2): while $\Delta > \theta$:
(S3):     for all $x \in \mathcal{S}$:
(S4):         $V_{new}(x) = \max_{a \in \mathcal{A}(x)} \left( r(x,a) + \beta \sum_{x' \in \mathcal{S}} P[x' \mid x, a] \, V(x') \right)$
(S5):     Compute $\Delta = \max_{x \in \mathcal{S}} |V_{k+1}(x) - V_k(x)|$.
(S6):     Set $V(x) = V_{new}(x)$ for all $x \in \mathcal{S}$.
(S7): For all $x \in \mathcal{S}$:
          $\pi^*(x) = \operatorname{argmax}_{a \in \mathcal{A}(x)} \left( r(x,a) + \beta \sum_{x' \in \mathcal{S}} P[x' \mid x, a] \, V(x') \right)$
(S8): Return $\pi^*$.
```

Policy Iteration

- Value iteration is one of the algorithms to compute the optimal policy for a MDP.

- We now look into another algorithm for Policy iteration (PI) that can be used to compute the optimal policy for a MDP.
  - In fact, we can show that **Value iteration is a special case of PI**.
  - Also, **PI is more intuitive** to understand if you don't have any background in Dynamic programming; VI was easier for you because you did Dynamic programming before (same may not be true for others).

Policy Iteration

- The overall idea behind Policy iteration (PI) are the following two steps:

Policy evaluation: Evaluate the performance of the current policy π using policy evaluation.

Policy update: Use the result of policy evaluation to choose a new policy π_{new} which is "better" then π. What is "better"? Well obviously, the value function of π_{new} has to be more than π.

After completing the policy update step, we go back to policy evaluation step with $\pi = \pi_{new}$ and the cycle continues until convergence.

Policy Iteration

- The overall idea behind Policy iteration (PI) are the following two steps:

Policy evaluation: Evaluate the performance of the current policy π using policy evaluation.

Policy update: Use the result of policy evaluation to choose a new policy π_{new} which is "better" then π. What is "better"? Well obviously, the value function of π_{new} has to be more than π.

After completing the policy update step, we go back to policy evaluation step with $\pi = \pi_{new}$ and the cycle continues until convergence.

- We have previously discussed in this slide (click here) that policy evaluation is a critical step to find the optimal policy, i.e. only when we know how a policy is performing can know how to improve it. This is in fact how human behaves and is also the essence of "learning". For this reason, **PI is more intuitive** than VI.

NOTE: Here the word "learning" DOES NOT mean that the involved probability distribution are not available.

Policy Iteration

- The overall idea behind Policy iteration (PI) are the following two steps:

Policy evaluation: Evaluate the performance of the current policy π using policy evaluation.

Policy update: **Use the result of policy evaluation to choose a new policy $\boldsymbol{\pi_{new}}$** which is "better" then π. What is "better"? Well obviously, the value function of π_{new} has to be more than π.

After completing the policy update step, we go back to policy evaluation step with $\pi = \pi_{new}$ and the cycle continues until convergence.

- We also used policy evaluation before in this slide (click here) to come up with a brute force algorithm that computes the optimal policy. But PI is better than this brute force algorithm because **PI uses the result of the policy evaluation step to SMARTLY choose a new policy $\boldsymbol{\pi_{new}}$** rather than simply iterating over all the policies like the brute force algorithm.

Policy Iteration: Policy Improvement Theorem

- A policy π_{new} is better than π if the value function corresponding to π_{new} is more than the value function corresponding to π for all the states. Mathematically,

$$V^{\pi_{new}}(x) \geq V^{\pi}(x)\,, \forall x \in \mathcal{S} \qquad \text{(E.20.)}$$

- Policy improvement theorem lies in the heart of choosing π_{new} from π that satisfies (E.20.).

Policy Improvement Theorem: Consider two policies π_1 and π_2. If,

$$q^{\pi_1}\big(x, \pi_2(x)\big) \geq V^{\pi_1}(x)\,, \forall x \in \mathcal{S} \qquad \text{(E.21.)}$$

Then,

$$V^{\pi_2}(x) \geq V^{\pi_1}(x)\,, \forall x \in \mathcal{S}$$

Proof: The formal proof is there in page number 78 of the book. We will first try to understand the intuition.

Policy Iteration: Policy Improvement Theorem

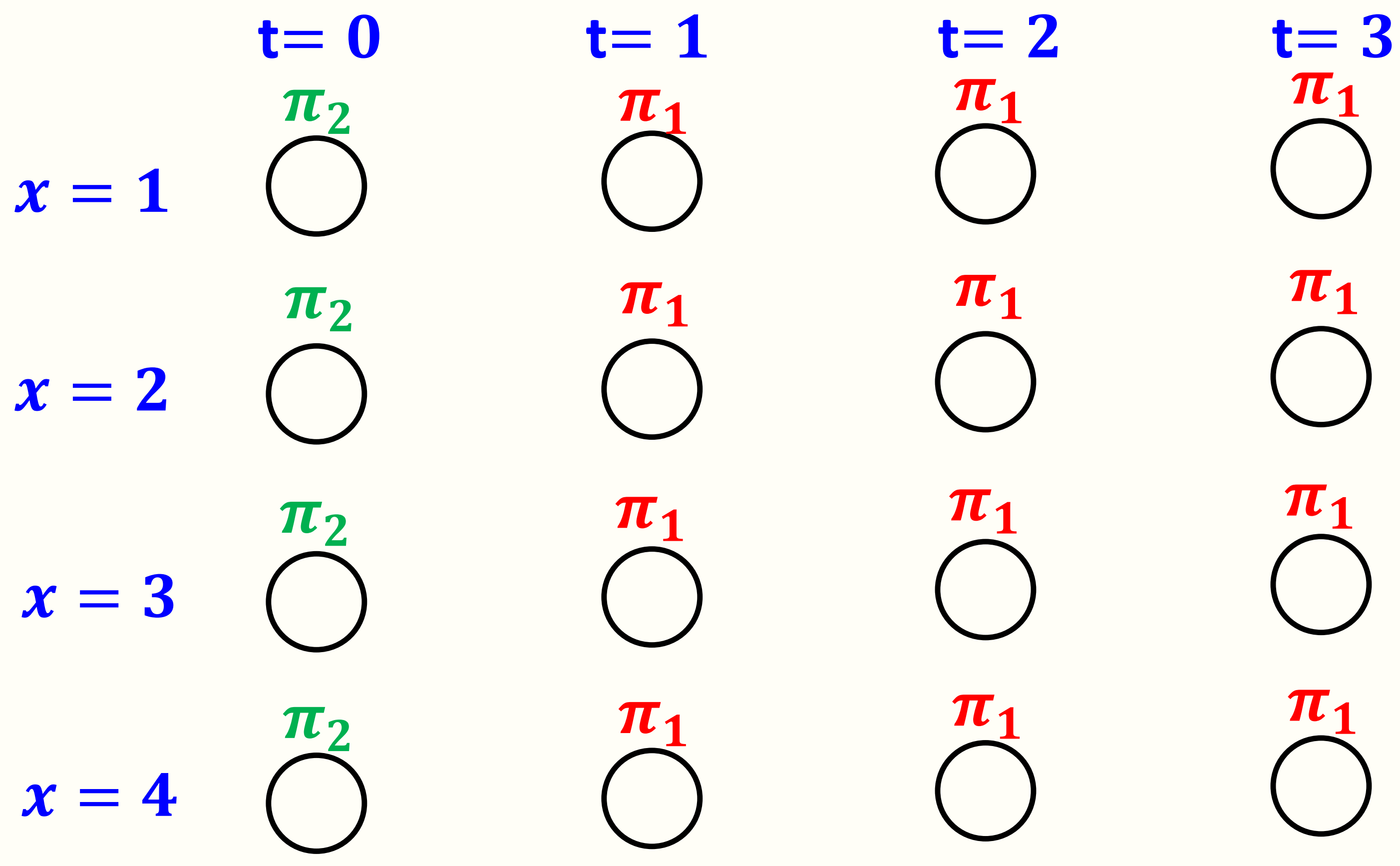
Policy Improvement Theorem: Consider two policies π_1 and π_2. If,

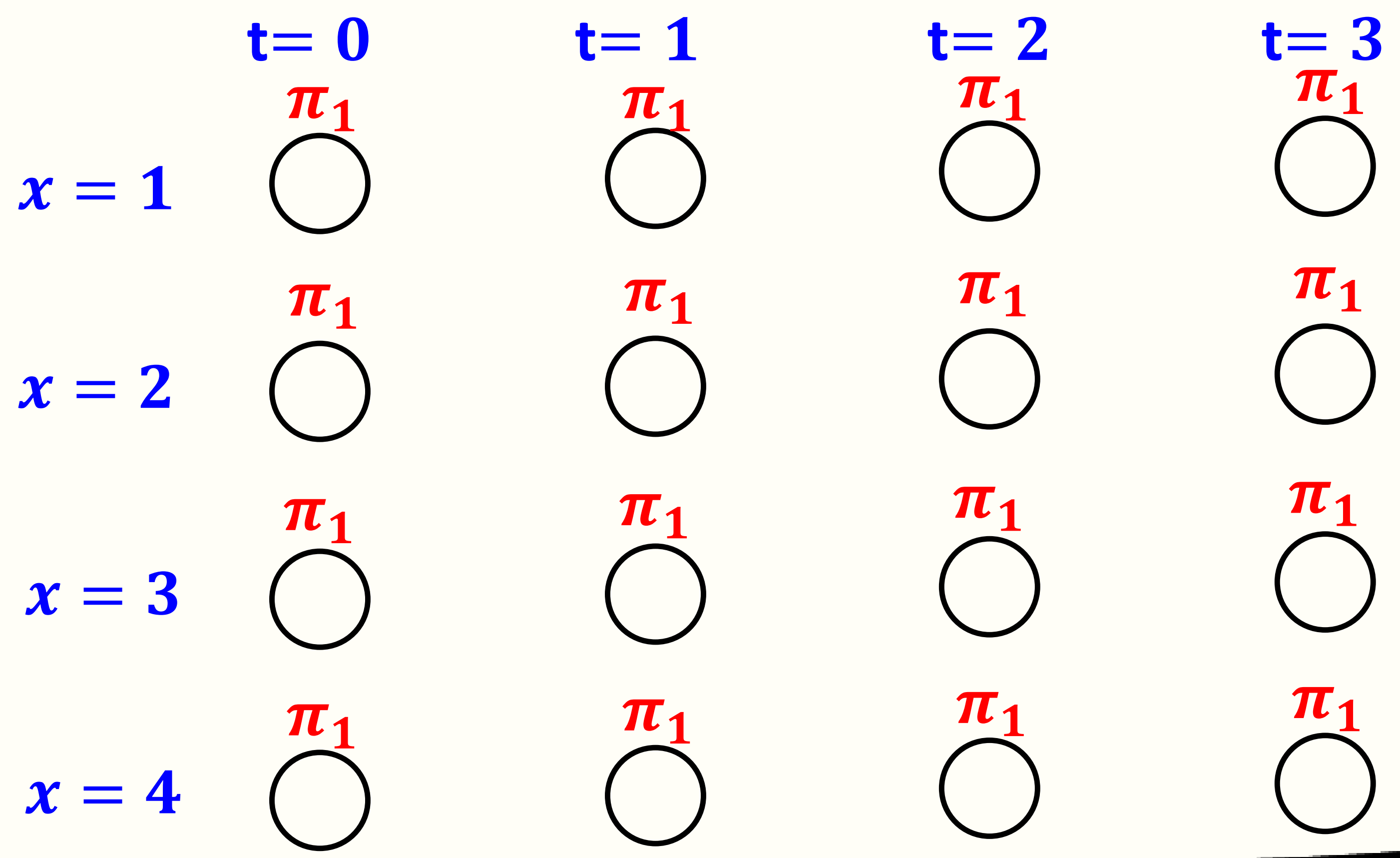
$$q^{\pi_1}\left(x, \pi_2(x)\right) \geq V^{\pi_1}(x)\,, \forall x \in \mathcal{S} \tag{E.21.}$$

Then,

$$V^{\pi_2}(x) \geq V^{\pi_1}(x)\,, \forall x \in \mathcal{S}$$

Intuition: $q^{\pi_1}\left(x, \pi_2(x)\right)$ is the expected value of the return is the action at the start time is taken according to policy $\pi_2$and then follow policy π_1 from the next time slot. $V^{\pi_1}(x)$ is the expected value of the return if we always use policy π_1 to take action.





Policy Iteration: Policy Improvement Theorem

- The overall idea behind Policy iteration (PI) are the following two steps:

Policy evaluation: Evaluate the performance of the current policy π using policy evaluation.

Policy update: **Use the result of policy evaluation to choose a new policy $\boldsymbol{\pi_{new}}$** which is "better" then π. What is "better"? Well obviously, the value function of π_{new} has to be more than π.

- According to policy improvement theorem if we can find a policy π_{new} such that,

$$q^{\pi}\left(x, \pi_{new}(x)\right) \geq V^{\pi}(x), \forall x \in \mathcal{S}$$

Then,

$$V^{\pi_{new}}(x) \geq V^{\pi}(x), \forall x \in \mathcal{S}$$

Implying that policy π_{new} is a "better" policy compared to π. What can be the new policy π_{new}?

Policy Iteration: Policy Improvement Theorem

- What can be the new policy π_{new} such that,

$$q^{\pi}\left(x, \pi_{new}(x)\right) \geq V^{\pi}(x)\,, \forall x \in \mathcal{S}$$

- One possible π_{new} is a policy that is **GREEDY with respect to the value function of the current policy $\boldsymbol{\pi}$**. Mathematically,

$$\pi_{new}(x) = \underset{a \in \mathcal{A}(x)}{\operatorname{argmax}} \left(r(x, a) + \beta \sum_{x' \in \mathcal{S}} P[x' \mid x, a] V^{\pi}(x') \right)$$

Policy Iteration: Policy Improvement Theorem

$$\pi_{new}(x) = \underset{a \in \mathcal{A}(x)}{\operatorname{argmax}} \left(r(x,a) + \beta \sum_{x' \in \mathcal{S}} P[x' \mid x, a] V^{\pi}(x') \right)$$

- Does π_{new} satisfy,

$$q^{\pi}\big(x, \pi_{new}(x)\big) \geq V^{\pi}(x) \,, \forall x \in \mathcal{S}$$

$$V^{\pi}(x) = \left(r\big(x, \pi(x)\big) + \beta \sum_{x' \in \mathcal{S}} P[x' \mid x, \pi(x)] V^{\pi}(x') \right)$$

$$q^{\pi}\big(x, \pi_{new}(x)\big) = \max_{a \in \mathcal{A}(x)} \left(r(x,a) + \beta \sum_{x' \in \mathcal{S}} P[x' \mid x, a] V^{\pi}(x') \right)$$

Policy Iteration: Policy Improvement Theorem

$$\pi_{new}(x) = \underset{a \in \mathcal{A}(x)}{\operatorname{argmax}} \left(r(x,a) + \beta \sum_{x' \in \mathcal{S}} P[x' \mid x, a] V^{\pi}(x') \right)$$

- Will policy iteration ever converge?
  - To answer this question, we have to first define what convergence means for policy iteration?
  - I little bad news and a small modification.

Policy Iteration: Policy Improvement Theorem

$$\pi_{new}(x) = \underset{a \in \mathcal{A}(x)}{\operatorname{argmax}} \left(r(x,a) + \beta \sum_{x' \in \mathcal{S}} P[x' \mid x, a] V^{\pi}(x') \right)$$

- Will be converged policy be optimal?

Policy Iteration: Psuedocode

Given: A threshold, θ.

```
(S1): For all $x \in \mathcal{S}$ arbitrarily initialize: (i) a policy $\pi(x)$ to any value in $\mathcal{A}(x)$, and (ii) a value
      function $V(x)$ to any real value. Also, set $converged = False$.
(S2): while not(converged):
(S3):     Set Use IPE to compute the value function $V^{\pi}(x)$ for all $x \in \mathcal{S}$ corresponding to current
          policy $\pi$. Set the initial value function of IPE as $V(x)$ and the conversion threshold as $\theta$.
(S4):     For all $x \in \mathcal{S}$:
(S5):         Compute,
              $\tilde{a} = \underset{a \in \mathcal{A}(x)}{\operatorname{argmax}} \left( r(x,a) + \beta \sum_{x' \in \mathcal{S}} P[x' \mid x,a] V^{\pi}(x') \right)$
              $q^{\pi}(x,\tilde{a}) = r(x,\tilde{a}) + \beta \sum_{x' \in \mathcal{S}} P[x' \mid x,\tilde{a}] V^{\pi}(x')$
(S6):         If $q^{\pi}(x,\tilde{a}) > V^{\pi}(x)$, then set $\pi_{new}(x) = \tilde{a}$. Else set $\pi_{new}(x) = \pi(x)$.
(S7):     If $\pi_{new}(x) = \pi(x)$ for all $x \in \mathcal{S}$, set $converged = True$.
(S8):     Set $\pi(x) = \pi_{new}(x)$ and $V(x) = V^{\pi}(x)$ for all $x \in \mathcal{S}$.
(S9): Return $\pi$.
```

Generalizedc Policy Iteration

Appendix 1: Rigorous Derivation of Bellman Equation

- The derivation of IPE and Bellman equation that we did before is not a rigorous derivation. We do a more mathematically rigorous derivation of IPE and Bellman equation here.

- We start with derivation of IPE.

Rigorous Derivation of IPE and Bellman Equation

- The derivation of IPE and Bellman equation that we did before is not a rigorous derivation. We do a more mathematically rigorous derivation of IPE and Bellman equation here.
- We start with derivation of IPE.
- We have,

$$V^{\pi}_{k+1,T}(x) = E_{\pi}\left[G_{k+1,T} \mid x_{T-k} = x\right]$$

Rigorous Derivation of IPE and Bellman Equation

- The derivation of IPE and Bellman equation that we did before is not a rigorous derivation. We do a more mathematically rigorous derivation of IPE and Bellman equation here.
- We start with derivation of IPE.
- We have,

$$V_{k+1,T}^{\pi}(x) = E_{\pi}\left[G_{k+1,T} \mid x_{T-k} = x\right]$$

$$= \sum_{a \in \mathcal{A}(x)} E_{\pi}\left[G_{k+1,T} \mid x_{T-k} = x, a_{T-k} = a\right] P\left[a_{T-k} = a \mid x_{T-k} = x\right] \Bigg\} \text{ Using Law of Total Expectation}$$

Rigorous Derivation of IPE and Bellman Equation

- The derivation of IPE and Bellman equation that we did before is not a rigorous derivation. We do a more mathematically rigorous derivation of IPE and Bellman equation here.
- We start with derivation of IPE.
- We have,

$$V^{\pi}_{k+1,T}(x) = E_{\pi}\left[G_{k+1,T} \mid x_{T-k} = x\right]$$

$$= \sum_{a \in \mathcal{A}(x)} E_{\pi}\left[G_{k+1,T} \mid x_{T-k} = x, a_{T-k} = a\right] P\left[a_{T-k} = a \mid x_{T-k} = x\right]$$

$$= \sum_{a \in \mathcal{A}(x)} q^{\pi}_{k+1,T}(x,a)\, \pi(a \mid x)$$

Using definition of action-value function in this slide.

The probability in the 2nd equation is nothing but the policy.

Rigorous Derivation of IPE and Bellman Equation

- The derivation of IPE and Bellman equation that we did before is not a rigorous derivation. We do a more mathematically rigorous derivation of IPE and Bellman equation here.
- We start with derivation of IPE.
- We have,

$$
\begin{aligned}
V_{k+1,T}^{\pi}(x) &= E_{\pi}\left[G_{k+1,T} \mid x_{T-k} = x\right] \\
&= \sum_{a \in \mathcal{A}(x)} E_{\pi}\left[G_{k+1,T} \mid x_{T-k} = x, a_{T-k} = a\right] P\left[a_{T-k} = a \mid x_{T-k} = x\right] \\
&= \sum_{a \in \mathcal{A}(x)} q_{k+1,T}^{\pi}(x,a)\, \pi(a \mid x)
\end{aligned}
$$

- So we have,

$$
V_{k+1,T}^{\pi}(x) = \sum_{a \in \mathcal{A}(x)} \pi(a \mid x)\, q_{k+1,T}^{\pi}(x,a) \qquad \text{(E.2.)}
$$

Rigorous Derivation of IPE and Bellman Equation

- Now we will derive an expression for the action-value function (Q-function), $q_{k,T}^{\pi}(x,a)$.

$q_{k+1,T}^{\pi}(x,a)$

$= E_{\pi}\left[G_{k+1,T} \mid x_{T-k} = x, a_{T-k} = a\right]$

Rigorous Derivation of IPE and Bellman Equation

- Now we will derive an expression for the action-value function (Q-function), $q^{\pi}_{k,T}(x,a)$.

$$q^{\pi}_{k+1,T}(x,a)$$

$$= E_{\pi}\left[G_{k+1,T} \,\middle|\, x_{T-k}=x, a_{T-k}=a\right]$$

$$= E_{\pi}\left[r_{T-k}+\beta r_{T-k+1}+\beta^2 r_{T-k+2}+\cdots+\beta^k r_T \,\middle|\, x_{T-k}=x, a_{T-k}=a\right]$$

Obtained by expanding $G_{k+1,T}$ using the definition in this slide.

Rigorous Derivation of IPE and Bellman Equation

- Now we will derive an expression for the action-value function (Q-function), $q_{k,T}^{\pi}(x,a)$.

$$q_{k+1,T}^{\pi}(x,a)$$

$$= E_{\pi}\left[G_{k+1,T} \mid x_{T-k}=x, a_{T-k}=a\right]$$

$$= E_{\pi}\left[r_{T-k}+\beta r_{T-k+1}+\beta^{2} r_{T-k+2}+\cdots+\beta^{k} r_{T} \mid x_{T-k}=x, a_{T-k}=a\right]$$

$$= E\left[r_{T-k} \mid x_{T-k}=x, a_{T-k}=a\right]+\beta E\left[r_{T-k+1}+\beta r_{T-k+2}+\cdots+\beta^{k-1} r_{T} \mid x_{T-k}=x, a_{T-k}=a\right]$$

Rigorous Derivation of IPE and Bellman Equation

- Now we will derive an expression for the action-value function (Q-function), $q^{\pi}_{k,T}(x,a)$.

$$q^{\pi}_{k+1,T}(x,a)$$

$$= E_{\pi}\left[G_{k+1,T} \mid x_{T-k}=x, a_{T-k}=a\right]$$

$$= E_{\pi}\left[r_{T-k}+\beta r_{T-k+1}+\beta^2 r_{T-k+2}+\cdots+\beta^k r_T \mid x_{T-k}=x, a_{T-k}=a\right]$$

$$= \boxed{E\left[r_{T-k} \mid x_{T-k}=x, a_{T-k}=a\right]} + \beta E\left[r_{T-k+1}+\beta r_{T-k+2}+\cdots+\beta^{k-1} r_T \mid x_{T-k}=x, a_{T-k}=a\right]$$

$$= \boxed{r(x,a)} + \beta E\left[r_{T-k+1}+\beta r_{T-k+2}+\cdots+\beta^{k-1} r_T \mid x_{T-k}=x, a_{T-k}=a\right]$$

The expected value in the 3rd equation is the expected value of the reward probability distribution corresponding to state, x, and action, a, which is equal to,

$$r(x,a) = \sum_{\tilde{r}} \tilde{r}\, P[\tilde{r} \mid x,a]$$

NOTE: Reward probability distribution is **stationary**. Hence, average reward $r(x,a)$ is not dependent on the time slot.

Rigorous Derivation of IPE and Bellman Equation

- Now we will derive an expression for the action-value function (Q-function), $q^{\pi}_{k,T}(x, a)$.

$$q^{\pi}_{k+1,T}(x,a)$$

$$= E_{\pi}\left[G_{k+1,T} \mid x_{T-k} = x, a_{T-k} = a\right]$$

$$= E_{\pi}\left[r_{T-k} + \beta r_{T-k+1} + \beta^2 r_{T-k+2} + \cdots + \beta^k r_T \mid x_{T-k} = x, a_{T-k} = a\right]$$

$$= E\left[r_{T-k} \mid x_{T-k} = x, a_{T-k} = a\right] + \beta E\left[r_{T-k+1} + \beta r_{T-k+2} + \cdots + \beta^{k-1} r_T \mid x_{T-k} = x, a_{T-k} = a\right]$$

$$= r(x,a) + \beta E\left[r_{T-k+1} + \beta r_{T-k+2} + \cdots + \beta^{k-1} r_T \mid x_{T-k} = x, a_{T-k} = a\right] \qquad \text{(E.3.)}$$

Rigorous Derivation of IPE and Bellman Equation

- Now we will derive an expression for the action-value function (Q-function), $q^{\pi}_{k,T}(x,a)$.

$$q^{\pi}_{k+1,T}(x,a)$$

$$= E_{\pi}\left[G_{k+1,T} \mid x_{T-k} = x, a_{T-k} = a\right]$$

$$= E_{\pi}\left[r_{T-k} + \beta r_{T-k+1} + \beta^2 r_{T-k+2} + \cdots + \beta^k r_T \mid x_{T-k} = x, a_{T-k} = a\right]$$

$$= E\left[r_{T-k} \mid x_{T-k} = x, a_{T-k} = a\right] + \beta E\left[r_{T-k+1} + \beta r_{T-k+2} + \cdots + \beta^{k-1} r_T \mid x_{T-k} = x, a_{T-k} = a\right]$$

$$= r(x,a) + \beta E\left[r_{T-k+1} + \beta r_{T-k+2} + \cdots + \beta^{k-1} r_T \mid x_{T-k} = x, a_{T-k} = a\right] \qquad \text{(E.3.)}$$

- Now we derive an expression for the second term of equation (E.3.).

Rigorous Derivation of IPE and Bellman Equation

- Now we will derive an expression for the action-value function (Q-function), $q^{\pi}_{k,T}(x, a)$.

$$E\left[r_{T-k+1} + \beta r_{T-k+2} + \cdots + \beta^{k-1} r_T \;\middle|\; x_{T-k} = x, a_{T-k} = a\right]$$

$$= \sum_{x' \in \mathcal{S}} E\left[r_{T-k+1} + \beta r_{T-k+2} + \cdots + \beta^{k-1} r_T \;\middle|\; x_{T-k} = x, a_{T-k} = a, x_{T-k+1} = x'\right] P\left[x_{T-k+1} = x' \;\middle|\; x_{T-k} = x, a_{T-k} = a\right]$$

Using Law of Total Expectation

Rigorous Derivation of IPE and Bellman Equation

➢ Now we will derive an expression for the action-value function (Q-function), $q^{\pi}_{k,T}(x,a)$.

$$E\left[r_{T-k+1} + \beta r_{T-k+2} + \cdots + \beta^{k-1} r_T \,\middle|\, x_{T-k} = x, a_{T-k} = a\right]$$

$$= \sum_{x' \in \mathcal{S}} \boxed{E\left[r_{T-k+1} + \beta r_{T-k+2} + \cdots + \beta^{k-1} r_T \,\middle|\, x_{T-k} = x, a_{T-k} = a, x_{T-k+1} = x'\right]} \boxed{P\left[x_{T-k+1} = x' \,\middle|\, x_{T-k} = x, a_{T-k} = a\right]}$$

$$= \sum_{x' \in \mathcal{S}} \boxed{E\left[r_{T-k+1} + \beta r_{T-k+2} + \cdots + \beta^{k-1} r_T \,\middle|\, x_{T-k+1} = x'\right]} \boxed{P\left[x' \,\middle|\, x, a\right]}$$

Using Markov property.

The term in the 2nd equation is just the state transition probability, i.e. the probability that the next state is x' given that the current state and current action are x and a respectively.

Rigorous Derivation of IPE and Bellman Equation

- Now we will derive an expression for the action-value function (Q-function), $q^{\pi}_{k,T}(x,a)$.

$$E\left[r_{T-k+1} + \beta r_{T-k+2} + \cdots + \beta^{k-1} r_T \;\middle|\; x_{T-k} = x, a_{T-k} = a\right]$$

$$= \sum_{x' \in \mathcal{S}} E\left[r_{T-k+1} + \beta r_{T-k+2} + \cdots + \beta^{k-1} r_T \;\middle|\; x_{T-k} = x, a_{T-k} = a, x_{T-k+1} = x'\right] P\left[x_{T-k+1} = x' \;\middle|\; x_{T-k} = x, a_{T-k} = a\right]$$

$$= \sum_{x' \in \mathcal{S}} E\left[r_{T-k+1} + \beta r_{T-k+2} + \cdots + \beta^{k-1} r_T \;\middle|\; x_{T-k+1} = x'\right] P\left[x' \;\middle|\; x, a\right]$$

$$= \sum_{x' \in \mathcal{S}} E\left[G_{k,T} \;\middle|\; x_{T-k+1} = x'\right] P\left[x' \;\middle|\; x, a\right]$$

Using definition of $G_{k,T}$ in this slide.

Rigorous Derivation of IPE and Bellman Equation

- Now we will derive an expression for the action-value function (Q-function), $q^{\pi}_{k,T}(x,a)$.

$$E\left[r_{T-k+1}+\beta r_{T-k+2}+\cdots+\beta^{k-1}r_T \,\middle|\, x_{T-k}=x, a_{T-k}=a\right]$$

$$=\sum_{x'\in\mathcal{S}} E\left[r_{T-k+1}+\beta r_{T-k+2}+\cdots+\beta^{k-1}r_T \,\middle|\, x_{T-k}=x, a_{T-k}=a, x_{T-k+1}=x'\right] P\left[x_{T-k+1}=x' \,\middle|\, x_{T-k}=x, a_{T-k}=a\right]$$

$$=\sum_{x'\in\mathcal{S}} E\left[r_{T-k+1}+\beta r_{T-k+2}+\cdots+\beta^{k-1}r_T \,\middle|\, x_{T-k+1}=x'\right] P\left[x' \,\middle|\, x,a\right]$$

$$=\sum_{x'\in\mathcal{S}} \boxed{E\left[G_{k,T} \,\middle|\, x_{T-k+1}=x'\right]} P\left[x' \,\middle|\, x,a\right]$$

$$=\sum_{x'\in\mathcal{S}} P\left[x' \,\middle|\, x,a\right] \boxed{V^{\pi}_{k,T}\left(x'\right)} \qquad \text{(E.4.)}$$

Using definition of $V^{\pi}_{k,T}(x')$ in this slide.

Rigorous Derivation of IPE and Bellman Equation

- Now, using equations (E.3.) and (E.4.) we get,

$$q^{\pi}_{k+1,T}\left(x,a\right)=r\left(x,a\right)+\beta\sum_{x^{'}\in\mathcal{S}}P\left[x^{'}\;\middle|\;x,a\right]V^{\pi}_{k,T}\left(x^{'}\right) \tag{E.5.}$$

- Using equations (E.2.) and (E.5.) we get,

$$V^{\pi}_{k+1,T}\left(x\right)=\sum_{a\in\mathcal{A}(x)}\pi\left(a\mid x\right)\left(r\left(x,a\right)+\beta\sum_{x^{'}\in\mathcal{S}}P\left[x^{'}\;\middle|\;x,a\right]V^{\pi}_{k,T}\left(x^{'}\right)\right) \tag{E.6.}$$

Rigorous Derivation of IPE and Bellman Equation

$$V^{\pi}_{k+1,T}(x) = \sum_{a\in\mathcal{A}(x)} \pi(a \mid x)\left(r(x,a) + \beta \sum_{x'\in\mathcal{S}} P\left[x' \,\middle|\, x,a\right] V^{\pi}_{k,T}\left(x'\right)\right) \quad \text{(E.6.)}$$

- As $T \to \infty$, $V^{\pi}_{k,T}(x) \to V^{\pi}(x)$ because of the stationarity and infinite series reason mentioned in this slide. $V^{\pi}_{k,T}(x) \to V^{\pi}(x)$ implies that the value function is not dependent on k.

  Consequently, as $T \to \infty$ equation (E.6.) becomes,

$$V^{\pi}(x) = \sum_{a\in\mathcal{A}(x)} \pi(a \mid x)\left(r(x,a) + \beta \sum_{x'\in\mathcal{S}} P\left[x' \,\middle|\, x,a\right] V^{\pi}\left(x'\right)\right) \quad \text{(E.7.)}$$

Equation (E.7.) is called the Bellman equation.

Thank you