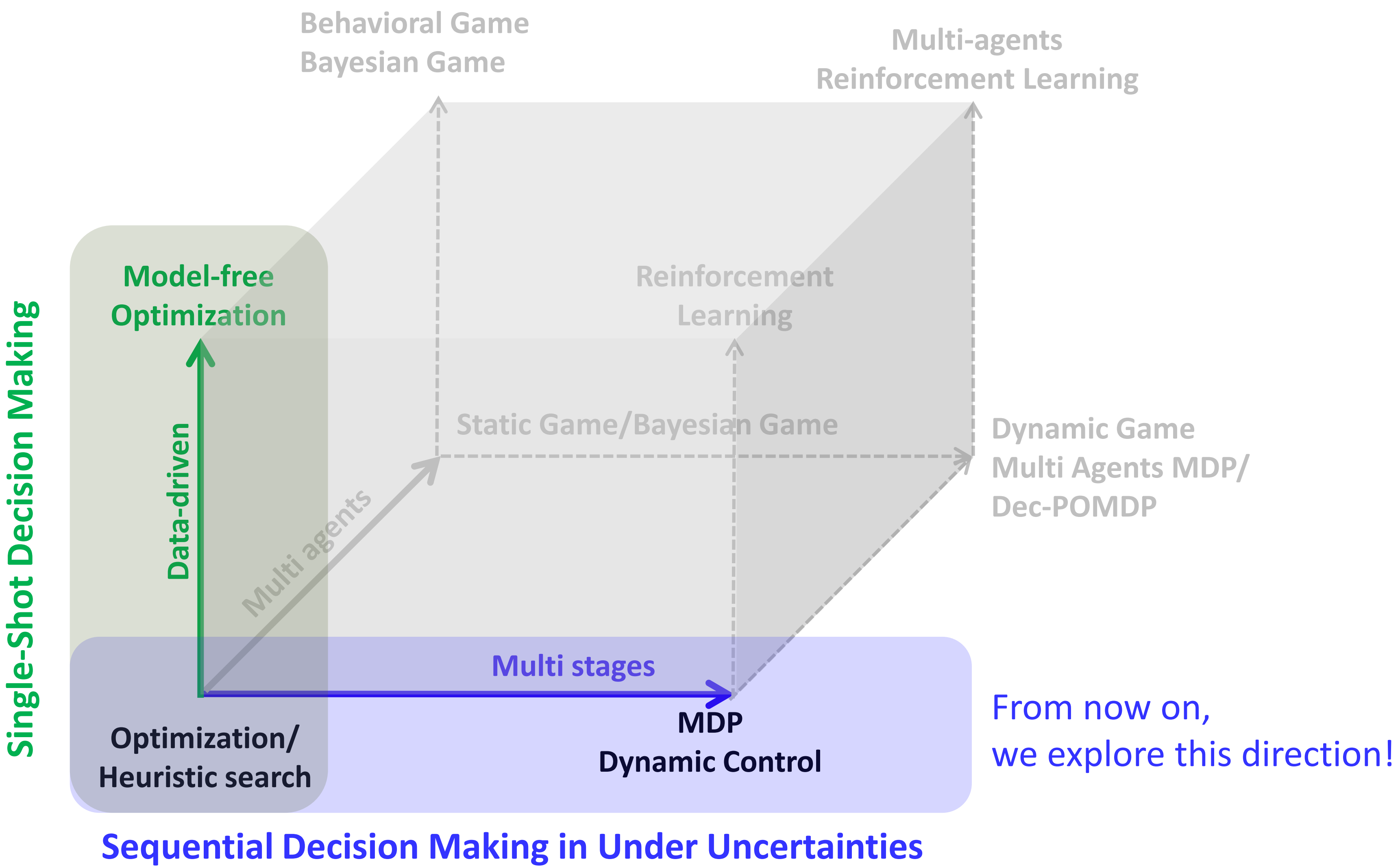
L13. Markov Decision Process (Formulation)

1. MDP definition
2. Cost (reward)
3. Transition probability



- Requires resonating about future *sequences* of actions and observations

Tension between breadth of applicability and mathematical tractability

Real world problems

Breadth of applicability ⟷ Mathematical tractability

Balance!

Sequential Decision Making in Uncertainties

- Many important problems require the decision maker to make a series of decisions.
- It requires resonating about future *sequences* of actions and observations

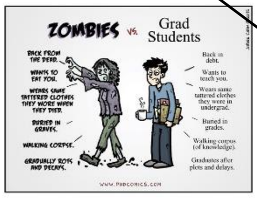
State:
Senior year

Action 1 : Get a job
Reward : 4,000$ per/mon

Action 2 : Go to graduate school
Reward : 500$ per/mon

Action 3 : Start up company
Reward : 0$ per/mon

State:



Action :
Publish X paper

No job

Researcher

Professor

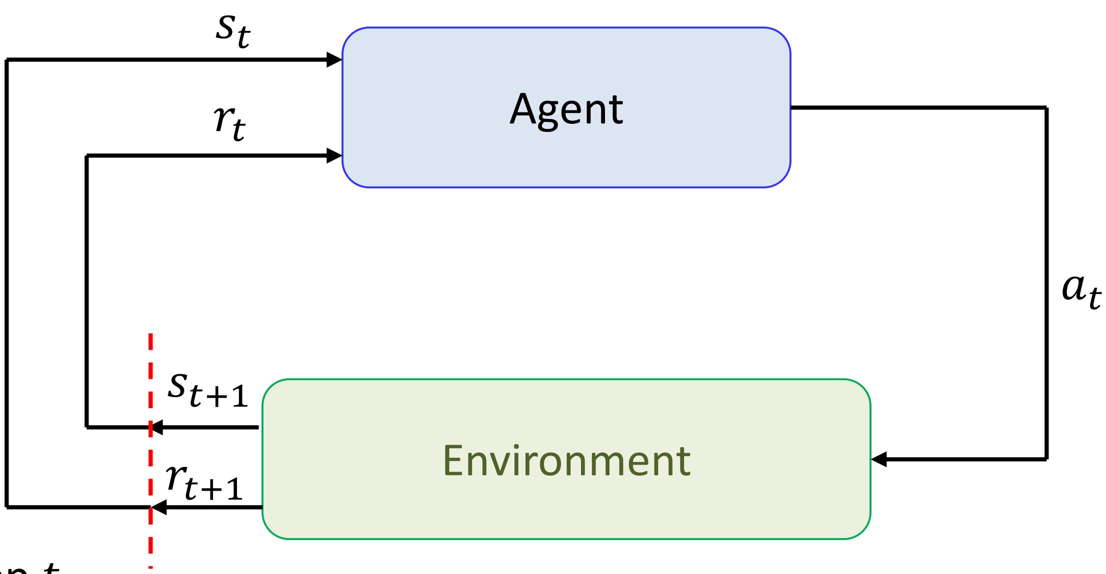
Private company

Officer

House keeper

- taking an action might lead to any one of many possible states
- how we can even hope to act optimally in the face of randomness?

The Agent-Environment Interface



At each time step t,

- The agent receives some representation of the environment's state $s_t \in \mathcal{S}$ and select an action $a_t \in \mathcal{A}(s_t)$
- One time step later, in part as a consequence of its action, the agent receives a numerical reward, $r_{t+1} \in \mathcal{R}$ and finds itself in a new state s_{t+1}

The goal is to find the optimum policy $a_t = \pi_t(s_t)$ mapping the current state s_t to the optimum action a_t^* to maximize the total amount of reward it receives over the long run

✓ Find policy π_t using analytical MDP models→ Dynamic programming (next week)
✓ Find policy π_t using data→ Reinforcement learning (2 weeks later)

The Agent-Environment Interface

- The time steps $t = 0{,}1, \ldots$ need not refer to a fixed interval of real time:
  - ✓ they can refer to arbitrary successive stages of decision–making and action

- The actions can be
  - ✓ Low-level controls, such as the voltages applied to motors of a robot arm
  - ✓ High-level decisions, such as whether or not to go graduate school

- The states can take a wide variety of forms
  - ✓ Can be completely determined by low-level sensations, such as sensor readings
  - ✓ Can be high-level and abstract, such as image or mental status

- The reward is a consequence of taking an action given a state
  - ✓ Can be specified according to the target tasks
  - ✓ Maximizing reward should results in achieving the goals of a task

In summary,

Actions can be any decision we want to learn how they affect rewards, and the states can be anything we can know that might be useful in making actions

The Boundary between Agent-Environment

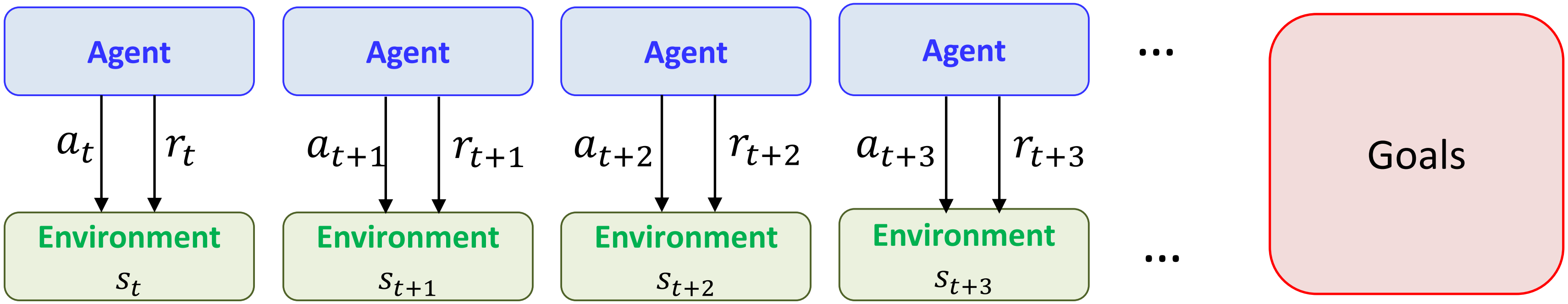
Agent :
Decision making algorithm

Environment:
Robot arm, motor, actuator, battery…

- Anything that cannot be changed arbitrarily by the agent is considered to be outside of the agent and thus part of its environment
- Reward computation should be outside of agent so that agent cannot arbitrarily change them

The agent-environment boundary represents the limit of the **agent's absolute control**, not of its knowledge

Goals and Rewards



- The agent's goal is to maximize the total amount of reward (cumulative reward) it receives.

$$\max_{a_t, a_{t+1}, \ldots} \Sigma_k r_{t+k}$$

- Rewards we setup truly indicate what we want to be accomplished
- The reward signal is your way of communicating to the agent what you want it to achieve, not how you want it achieved

$$\max_{actions} \sum_{t=now}^{\text{End of life}} \underset{r_t}{\text{Annual Income}_t} \quad \xrightarrow{\quad ? \quad}$$

The goal of life: Happy life

Utility (returns)

How to formally define the accumulated reward in the long run?

- Denote reward : $r_t = R(S_t, A_t)$
- In the episodic tasks, the simplest utility is defined as

$$U_t = r_t + r_{t+1} + r_{t+3} + \cdots + r_T = \sum_{k=0}^{T} r_{t+k}$$

  - ✓ T is a final time step associated with the terminal time step T
  - ✓ Examples include, maze, go, chess, etc.

- In the continuing tasks, the discounted utility is defined as

$$U_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k r_{t+k}$$

  - ✓ γ is a parameter, $0 \leq \gamma \leq 1$, called discount rate
  - ✓ Examples include, maze game, Go, Chess, etc.
  - ✓ $\gamma = 0$ (myopic): concern only with maximizing immediate rewards
  - ✓ $\gamma = 1$ (farsighted): the objective takes future rewards take into account more strongly

Utility (returns)

Assumptions:

1. Remove index for episode number

Episode 1 $(s_{1,1}, a_{1,1}, r_{1.1}), (s_{2,1}, a_{2,1}, r_{2.1}),..., (s_{t,1}, a_{t,1}, r_{t.1}), \dots, (s_{T,1}, a_{T,1}, r_{T.1})$

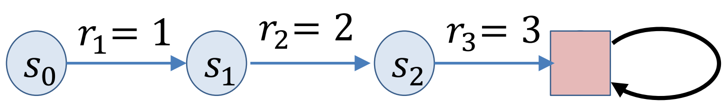
Episode 2 $(s_{1,2}, a_{1,2}, r_{1.2}), (s_{2,2}, a_{2,2}, r_{2.2}),..., (s_{t,n}, a_{t,n}, r_{t.2}), \dots, (s_{T,1}, a_{T,1}, r_{T.2})$

$\vdots \quad \vdots \quad \vdots \quad \vdots$

Episode n $(s_{1,n}, a_{1,n}, r_{1.n}), (s_{2,n}, a_{2,n}, r_{2.n}),..., (s_{t,n}, a_{t,n}, r_{t,n}), \dots, (s_{T,n}, a_{T,n}, r_{T.n})$

$(s_1, a_1, r_1), (s_2, a_2, r_2),..., (s_t, a_t, r_t), \dots, (s_T, a_T, r_T)$

2. Introduce the observing state, in which agent transits the same state with no reward



$r_4 = 0,\ r_5 = 0,\ r_6 = 0, \dots$

- In the continuing tasks + the episodic tacks , the unified utility is defined as

$$U_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k r_{t+k}$$

✓ γ is a parameter, $0 \le \gamma \le 1$, called discount rate

✓ Examples include, maze game, Go, Chess, etc.

✓ $\gamma = 0$ (myopic): concern only with maximizing immediate rewards

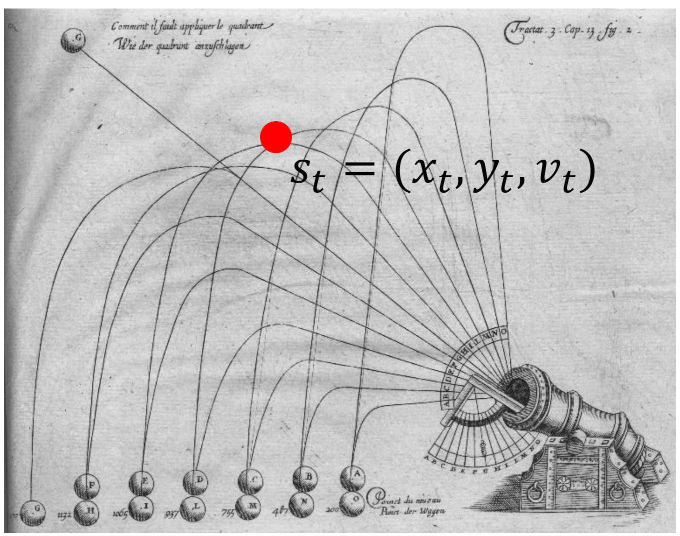
✓ $\gamma = 1$ (farsighted): the objective takes future rewards take into account more strongly

The Markov Property

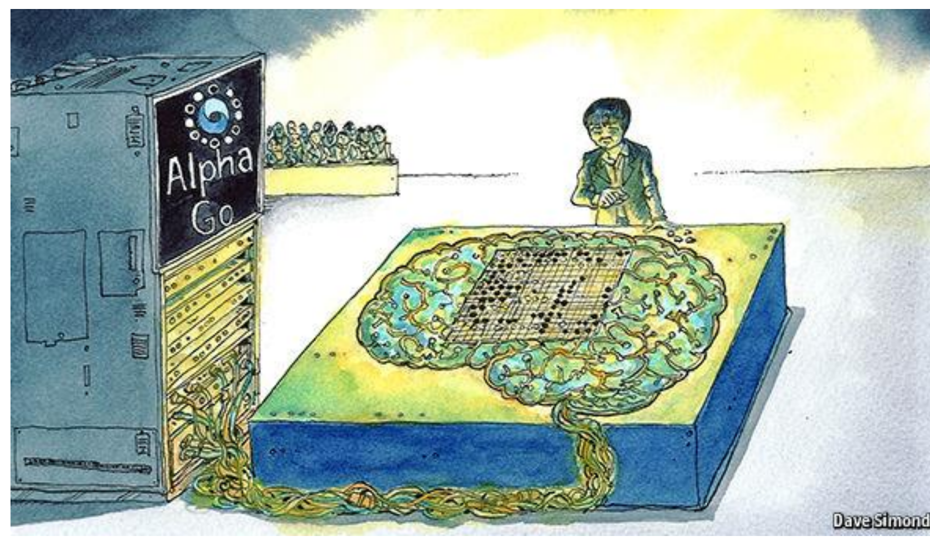
- The state is constructed and maintained on the basis of immediate sensations together with the previous state or some other memory of past sensations

- Ideal state signal summarizes past sensations compactly, yet in such a way that all relevant information is retained

- A state signal that succeeds in retaining all relevant information is said to be *Markov*



s_t = the location and velocity of the flying canon



s_t = the configuration of the black and white stones

The Markov Property

- "Markov" generally means that given the present state, the future and the past are independent

Andrey Markov (1856-1922)

- For Markov decision processes, "Markov" means action outcomes depend only on the current state

$$P(S_{t+1} = s' | S_t = s_t, A_t = a_t, S_{t-1} = s_{t-1}, A_{t-1} = a_{t-1}, \dots S_0 = s_0, A_0 = a_0)$$
$$= P(S_{t+1} = s' | S_t = s_t, A_t = a_t,)$$

- The best policy for choosing actions as a function of a Markov state is just as good as the best policy for choosing actions as a function of complete history

$$\pi^*(s_t, s_{t-1}, \dots, s_0) = \pi^*(s_t)$$

Note:

- As states become more Markovian, the better performance in MDP solution
- It is useful to think of the state at each time step as an approximation to a Markov value

Policy

Policy Reward

- A policy π gives an action $a \in \mathcal{A}$ for each state s $\in \mathcal{S}$:

$$\pi: \mathcal{S} \rightarrow \mathcal{A}$$

- An optimal policy π^* is one that maximizes expected utility

$$\pi^* = \underset{\pi}{\text{argmax}}\, \mathbb{E}[U^\pi(\mathrm{s})] \text{, for all s}$$

$$\text{where } \mathbb{E}[U^\pi(\mathrm{s})] = \mathbb{E}[r_t + \gamma r_{t+1} + \gamma^2 r_{t+3} + \cdots]$$

- An optimal policy can be either deterministic or stochastic

deterministic

$$a^* = \pi^*(s)$$

Take action a^* with a probability one

Stochastic

$$p(a|s) = \pi^*(s, a)$$

Take action a with a probability $p(a|s)$

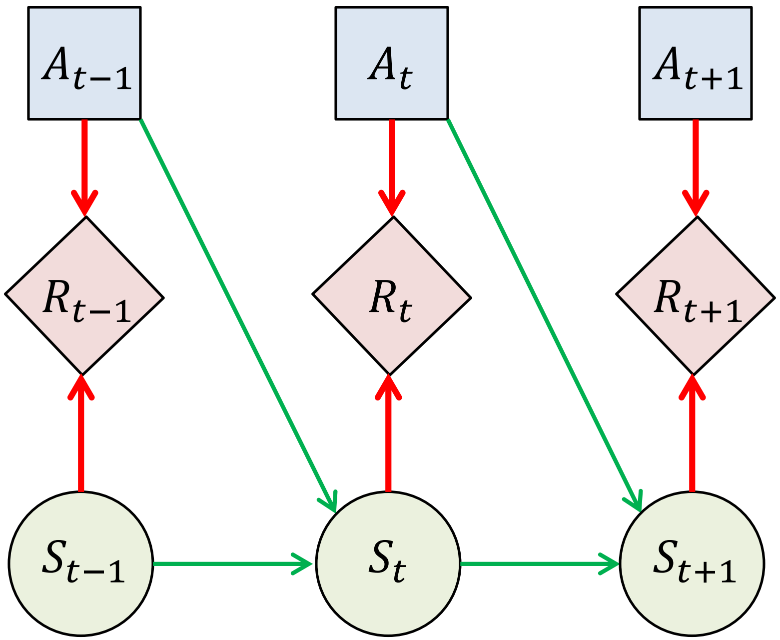
We will exclusively consider deterministic policy

Finite Markov Decision Process (MDP) :The state and action space are finite

An MDP is defined by:

- A set of states $s \in \mathcal{S}$
- A set of actions $a \in \mathcal{A}$
- A transition function $T(s, a, s') = P(S_{t+1} = s' | S_t = s, A_t = a) = P(s' | s, a)$
  - ✓ Probability that a state change occurs from s to s' when taking action a
  - ✓ Also called the model or the dynamics
- A reward function $R(s, a, s')$
  - ✓ $r_t = R(s_t, a_t, s_{t+1})$ or $r_{t+1} = R(s_t, a_t)$
  - ✓ If stochastic, $R(s, a, s') = \mathbb{E}[r_t + r_{t+1}, \ldots | S_t = s, A_t = a, S_{t+1} = s']$
- A start state $s_0 \in \mathcal{S}$
- A terminal state $s_T \in \mathcal{S}_T$ (for episodic tasks, and the terminal state space $\mathcal{S}_T$ is usually different from any other states space $\mathcal{S}$)

Markov Decision Processes



A_t : action taken at time t

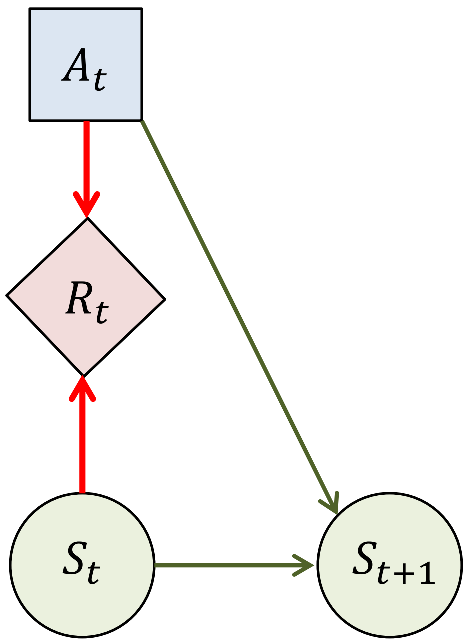
R_t : reward received at time t

S_t : state at time t

- Transition probability $T_t(s_t, a_t, s_{t+1}) = P(S_{t+1} = s_{t+1} | S_t = s_t, A_t = a_t) = P(s_{t+1} | s_t, a_t)$
  - ✓ represents the probability of transitioning from state s_t to s_{t+1} after executing action s_t at time t (Markov assumption)

- Reward function: $r_t = R_t(s_t, a_t)$: represents the candidates of reward values when executing action a_t from state s_t at time t

(Stationary) Markov Decision Processes



Stationary Markov Decision Process (MDP)

→ transition and reward models are stationary

- Transition probability $T(s_t, a_t, s_{t+1})$ are the same for all time step t
- Reward function: $r_t = R(s_t, a_t)$ are the same for all time step t

Value Function & Q function

Value function (state value function for π)

"How good it is for the agent to be in a given state"

$V^{\pi}(s)$: The expected utility received by following policy π from state s

$$V^{\pi}(s) = \mathbb{E}_{\pi}(U_t | S_t = s) = \mathbb{E}_{\pi}\left(\sum_{k=0}^{\infty} \gamma^k r_{t+k} \, | S_t = s\right)$$

$\mathbb{E}_{\pi}$: not expectation over policy π but all stochastic state transitions associated with π

Q-function (action-value function for π)

"How good it is for the agent to perform a given action in a given state"

$Q^{\pi}(s, a)$: The expected utility of taking action a from state s, and then following policy π

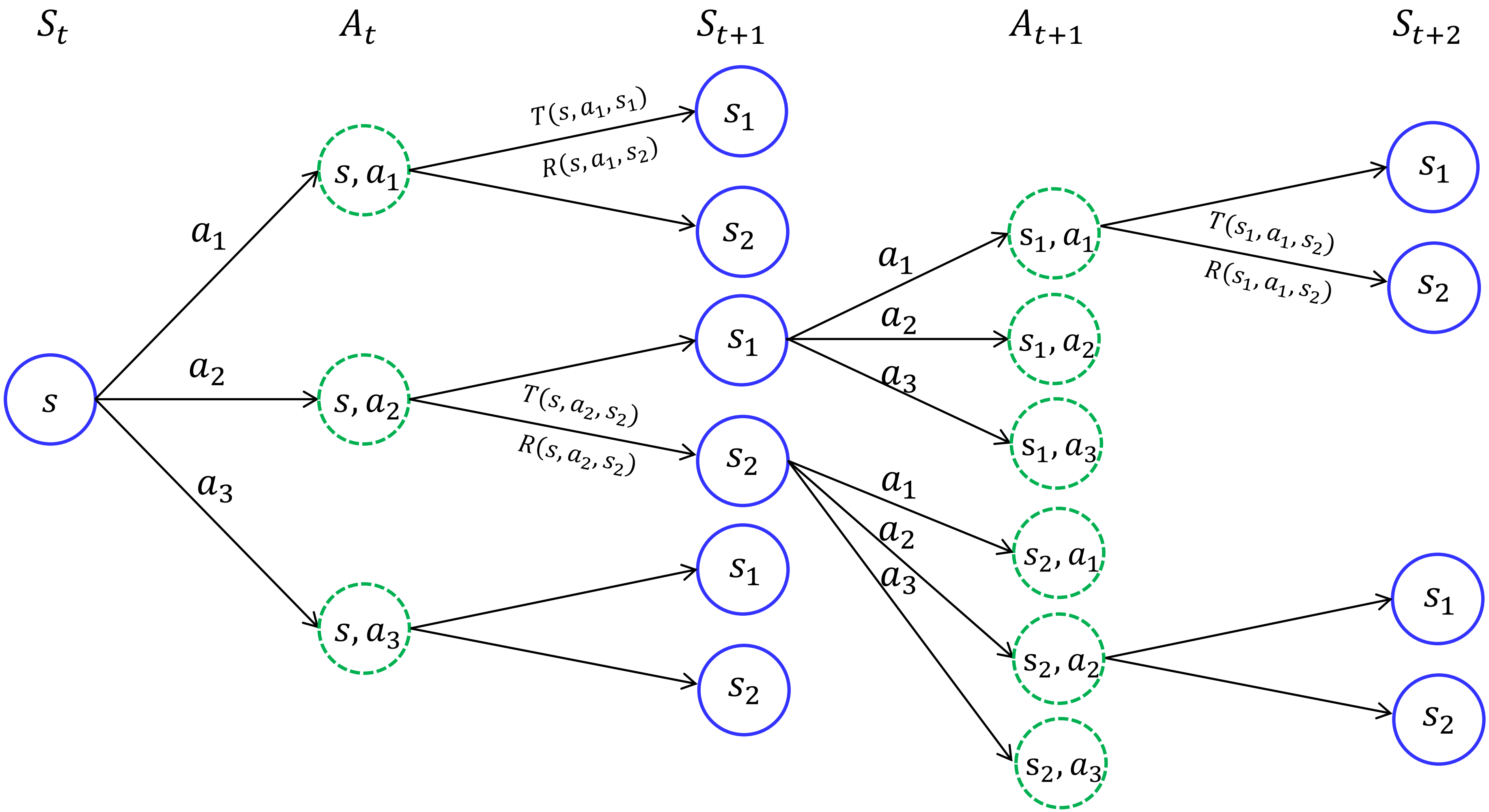
$$Q^{\pi}(s, a) = \mathbb{E}_{\pi}(U_t | S_t = s, A_t = a) = \mathbb{E}_{\pi}\left(\sum_{k=0}^{\infty} \gamma^k r_{t+k} \, | S_t = s, A_t = a\right)$$

Because the agent can expect to receive in the future depend on what actions it will take
→ Value and Q functions are defined with respect to a particular policy mapping state $s \in \mathcal{S}$ to an action $a \in \mathcal{A}$ (π is given.)

Value Function & Q function

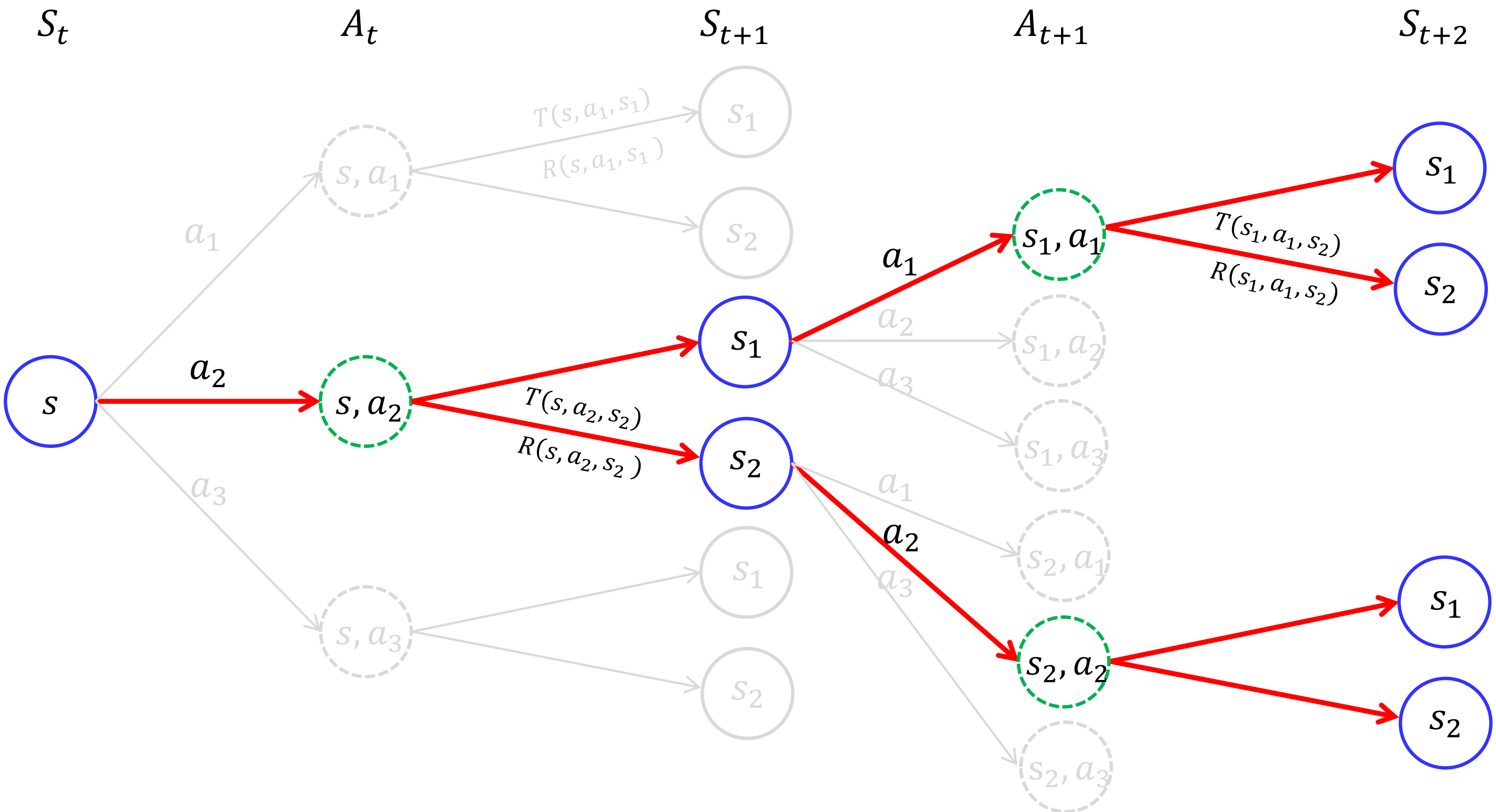
Value Function

All possible trajectories



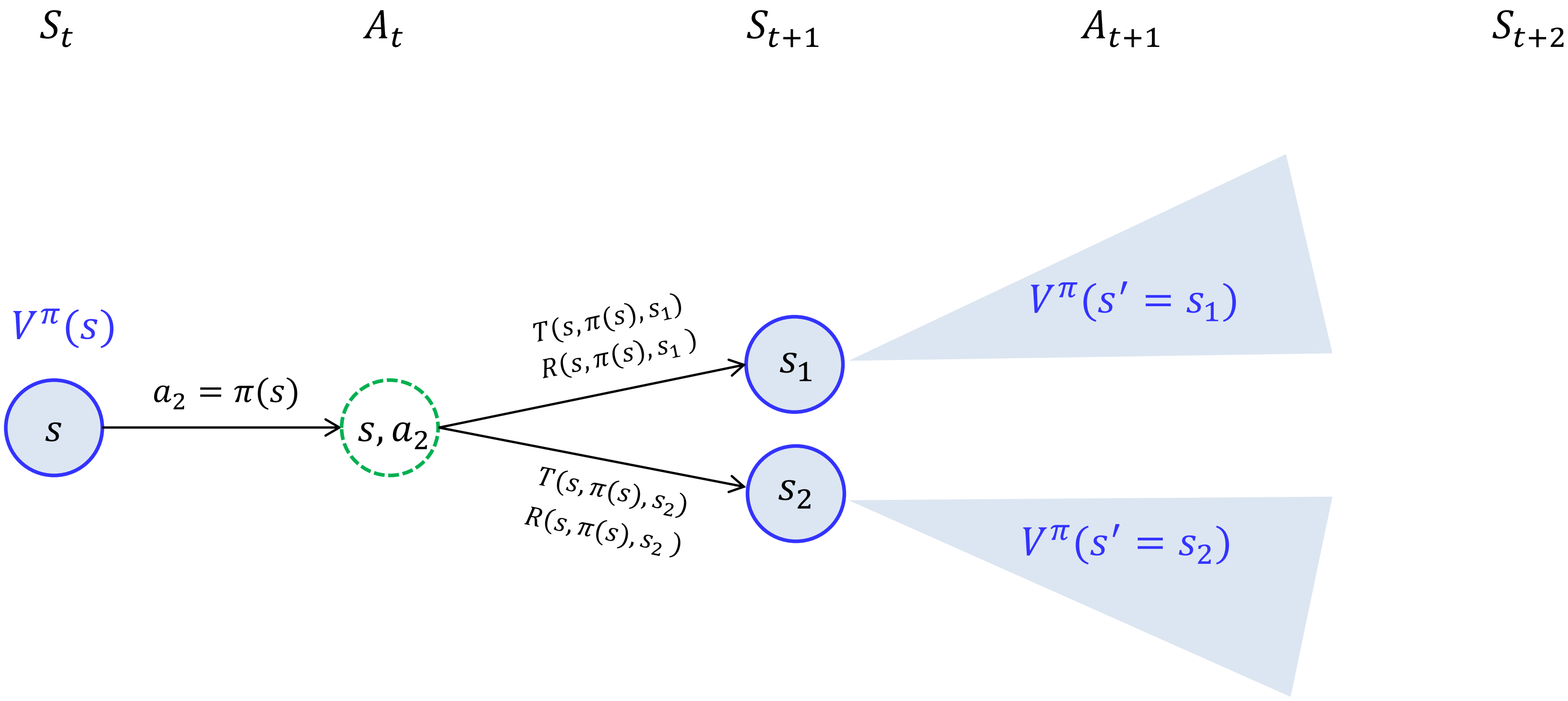
Value Function

A policy π is given as: $\pi(s) = a_2$; $\pi(s_1) = a_1$; $\pi(s_2) = a_2$



Value Function

A policy π is given as: $\pi(s) = a_2$; $\pi(s_1) = a_1$; $\pi(s_2) = a_2$



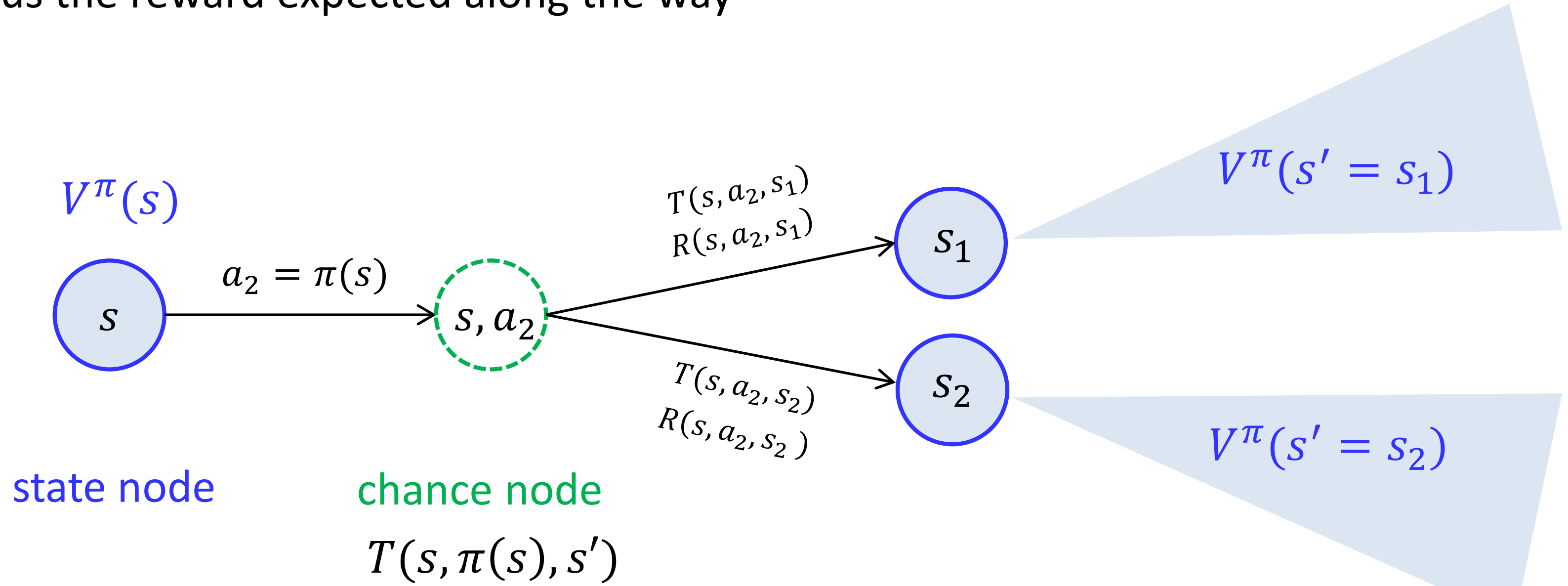
$$V^\pi(s) = T(s, \pi(s), s_1)\{R(s, \pi(s), s_1) + \gamma V^\pi(s_1)\} + T(s, \pi(s), s_2)\{R(s, \pi(s), s_2) + \gamma V^\pi(s_2)\}$$

The Bellman Equation for Value Function

Recursive Formulation (The Bellman equation)

$$
\begin{aligned}
V^{\pi}(s) &= \mathbb{E}_{\pi}\left(\sum_{k=0}^{\infty} \gamma^k r_{t+k} \mid S_t = s\right) \\
&= \mathbb{E}_{\pi}\left(r_t + \gamma \sum_{k=0}^{\infty} \gamma^k r_{t+1+k} \mid S_t = s\right) \\
&= \sum_{s'} T(s, \pi(s), s')\{R(s, \pi(s), s') + \gamma \mathbb{E}_{\pi}\left(\sum_{k=0}^{\infty} \gamma^k r_{t+1+k} \mid S_{t+1} = s'\right)\} \\
&= \sum_{s'} T(s, \pi(s), s')\{R(s, \pi(s), s') + \gamma V^{\pi}(s')\}
\end{aligned}
$$

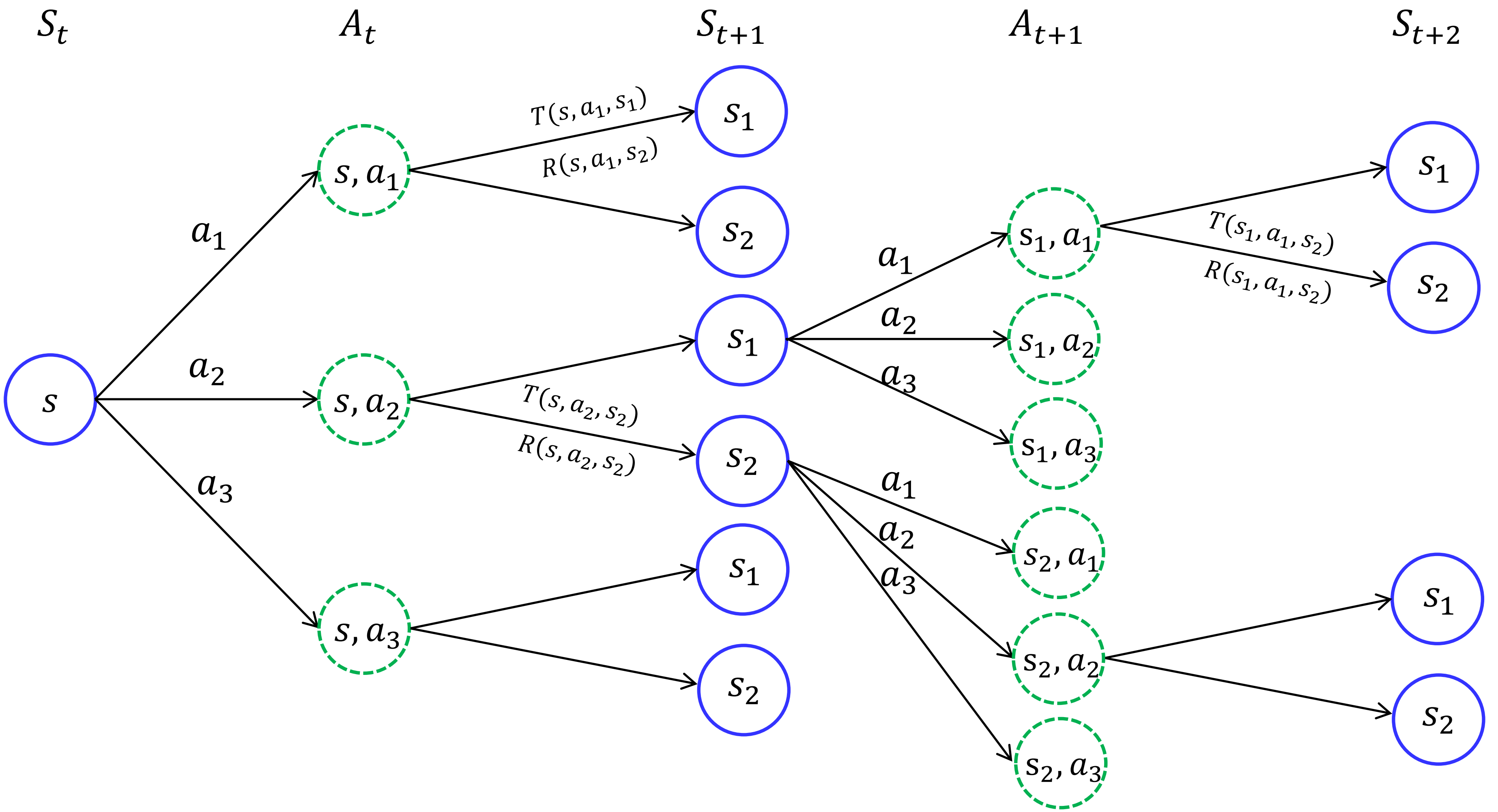
The value of the start state must equal the (discounted) value of the expected next state, plus the reward expected along the way



Value Function & Q function

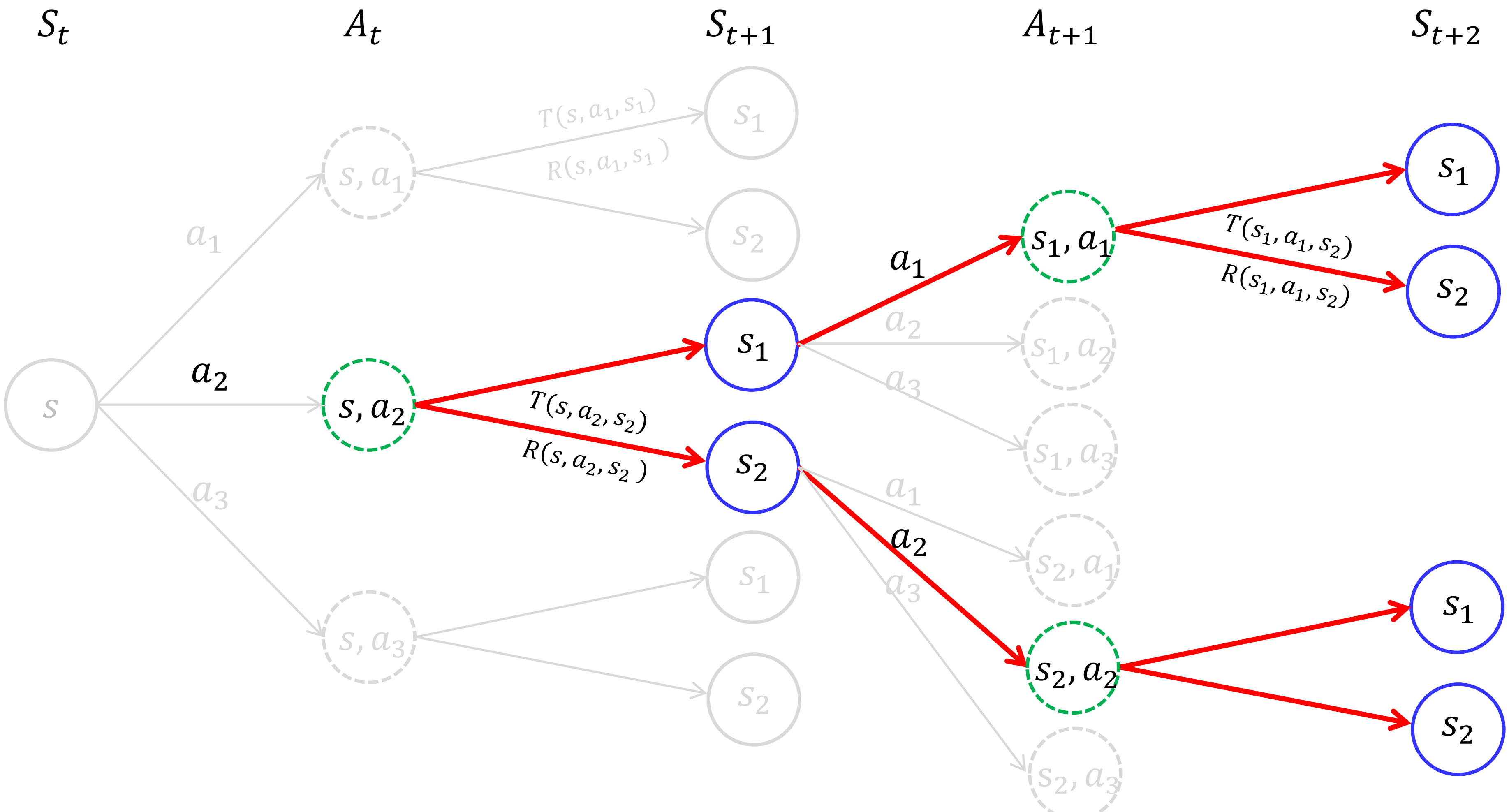
Q function

All possible trajectories



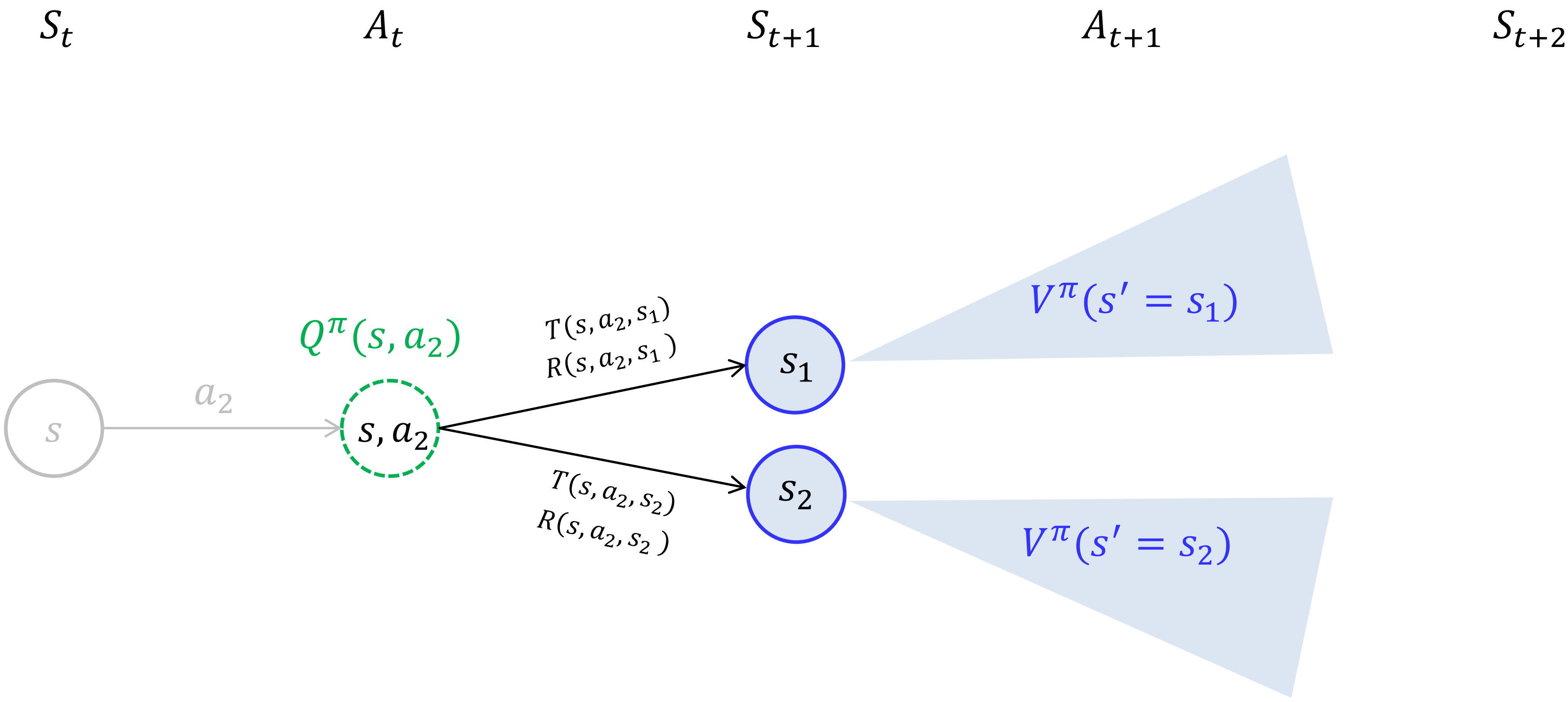
Q function

A policy π is given as: $\pi(s) = a_2$; $\pi(s_1) = a_1$; $\pi(s_2) = a_2$



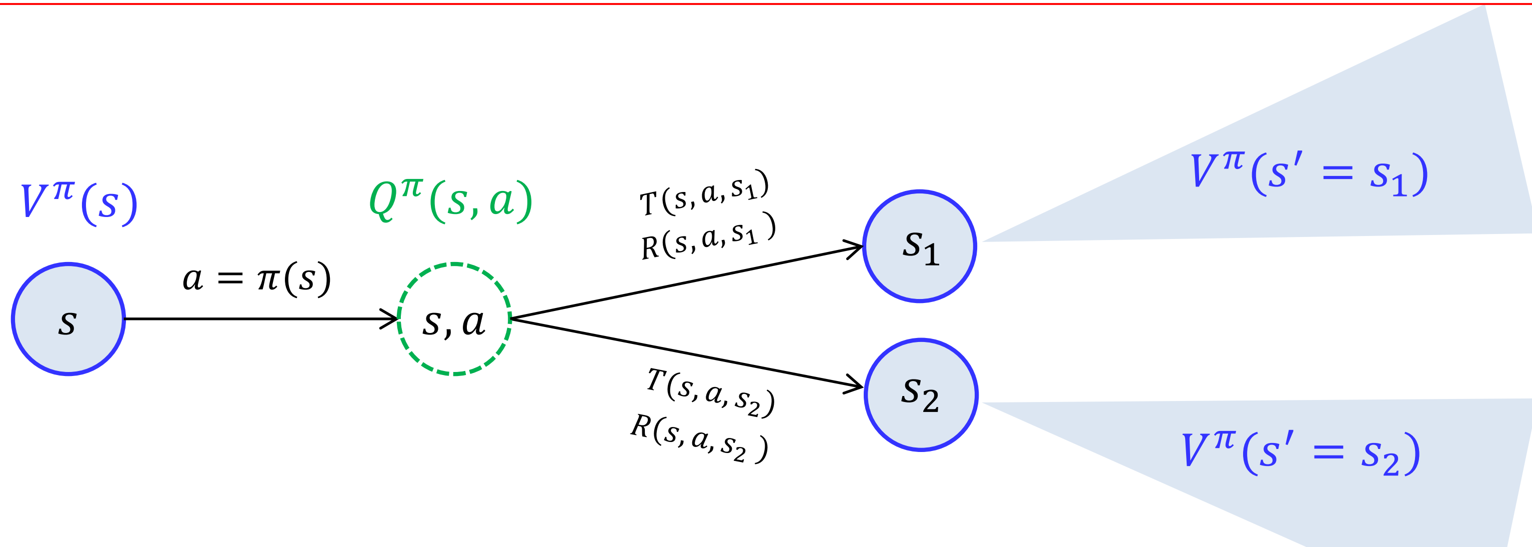
Q Function

A policy π is given as: $\pi(s) = a_2$; $\pi(s_1) = a_1$; $\pi(s_2) = a_2$



$$Q^\pi(s, a_2) = T(s, a_2, s_1)\{R(s, a_2, s_1) + \gamma V^\pi(s_1)\} + T(s, a_2, s_2)\{R(s, a_2, s_2) + \gamma V^\pi(s_2)\}$$

Q Function



$$Q^\pi(s,a) = \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma V^\pi(s')]$$

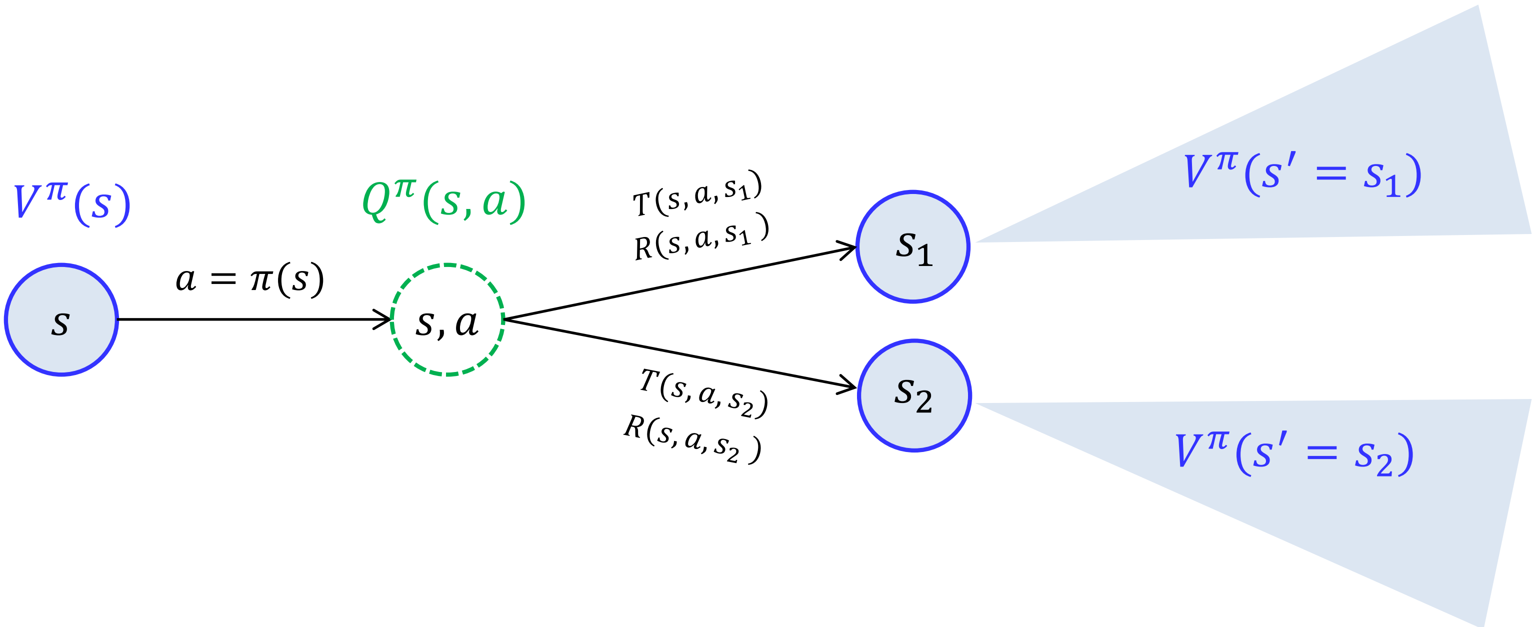
Note that:

$$V^\pi(s) = \sum_{s'} T(s,\pi(s),s')\{R(s,\pi(s),s') + \gamma V^\pi(s')\}$$

$$= Q^\pi(s,\pi(s))$$

- $Q^\pi(s,a)$ is more general since it has the option to select an action a given state s
- If the action is enforced to select a $= \pi(s)$ according to the policy π, $V^\pi(s) = Q^\pi(s,\pi(s))$

Summary for Value function and Q function



$$V^{\pi}(s) = \sum_{s'} T(s,\pi(s),s')\{R(s,\pi(s),s') + \gamma V^{\pi}(s')\}$$

$$Q^{\pi}(s,a) = \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma V^{\pi}(s')]$$

$$V^{\pi}(s) = Q^{\pi}(s,\pi(s))$$

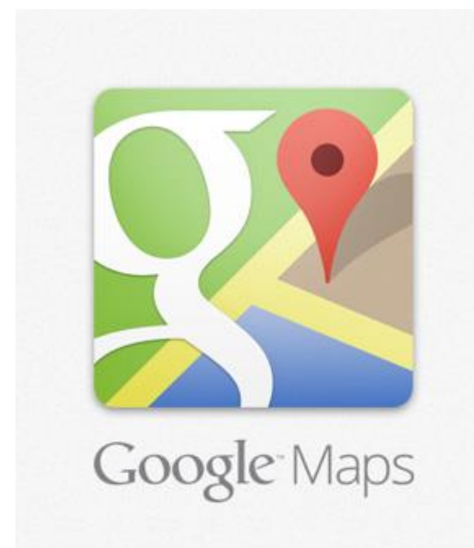
Bellman optimality equation for $V^*(s)$

Optimal policy

$$\pi^* \geq \pi \text{ if and only if } V^{\pi^*}(s) \geq V^{\pi}(s) \text{ for all } s$$

Optimal state- value function

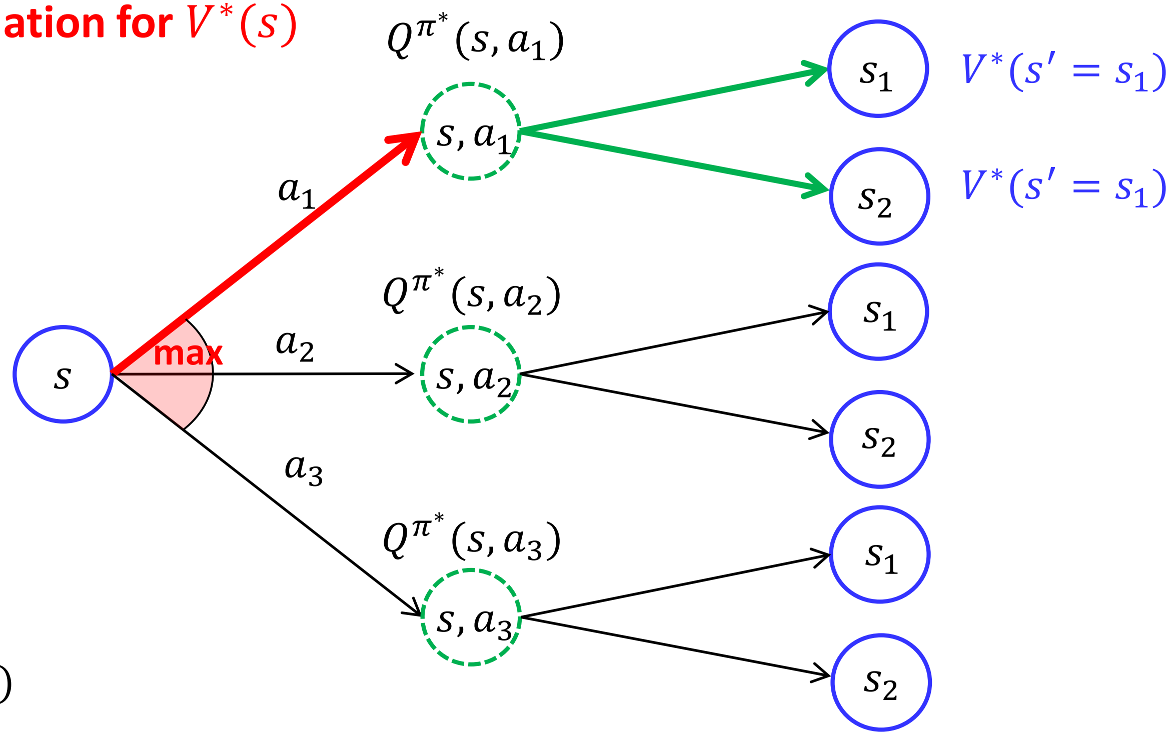
$$V^*(s) = \max_{\pi} V^{\pi}(s) \text{ , for all } s$$



Bellman Optimality Equation for State-Value Function

Bellman optimality equation for $V^*(s)$

$V^*(s) = V^{\pi^*}(s)$



$$V^{\pi^*}(s) = \max_{a \in \mathcal{A}(s)} Q^{\pi^*}(s, a)$$

$$= \max_{a \in \mathcal{A}(s)} \mathbb{E}_{\pi^*}\left(\sum_{k=0}^{\infty} \gamma^k r_{t+k} \,\middle|\, S_t = s, A_t = a\right)$$

$$= \max_{a \in \mathcal{A}(s)} \mathbb{E}_{\pi^*}\left(r_t + \gamma \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \,\middle|\, S_t = s, A_t = a\right)$$

$\mathbb{E}$ is over transitions associated with π^*

$$= \max_{a \in \mathcal{A}(s)} \mathbb{E}\left(r_t + \gamma V^{\pi^*}(s_{t+1}) \,\middle|\, S_t = s, A_t = a\right)$$

$\mathbb{E}$ is over transitions

$$= \max_{a \in \mathcal{A}(s)} \sum_{s'} T(s, a, s')\{R(s, a, s') + \gamma V^{\pi^*}(s')\}$$

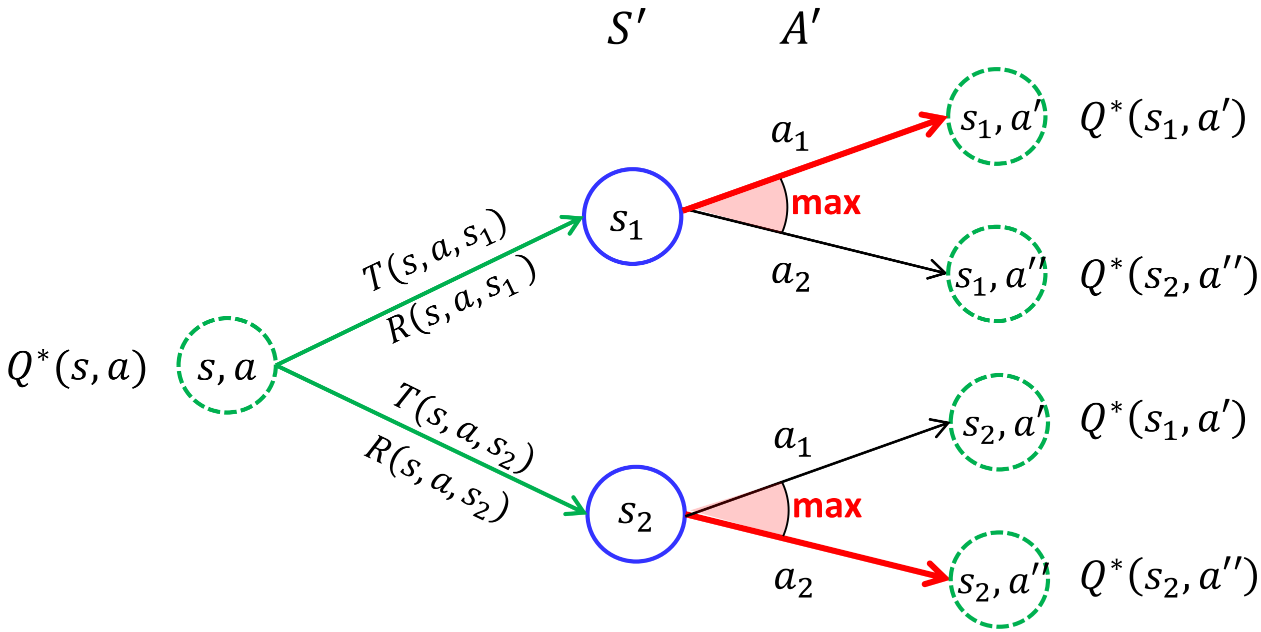
First take optimum action and follow the optimum policy

Bellman optimality equation for $Q^*(s,a)$

$Q^*(s,a) = Q^{\pi^*}(s,a) = \max_{\pi} Q^{\pi}(s,a)$, for all $s \in \mathcal{S}$ and $a \in \mathcal{A}(s)$

Bellman Optimality Equation for State-Action Value Function

Bellman optimality equation for $Q^*(s,a)$



$$Q^*(s,a) = \mathbb{E}\left\{r_t + \gamma \max_{a'} Q^*(s',a') | s_t = s, a_t = a\right\}$$
$$= \sum_{s'} T(s,a,s')\left\{R(s,a,s') + \gamma \max_{a'} Q^*(s',a')\right\}$$

$\mathbb{E}$ is over transitions $s \rightarrow s'$

First transits by transition probability and take the optimum action for each consequent states

Optimal Value Function & Q function

Relationships between $Q^*(s,a)$ and $V^*(s)$

$Q^*(s,a) = \max_\pi Q^\pi(s,a)$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}(s)$

$$\begin{aligned} Q^*(s,a) &= \max_\pi Q^\pi(s,a) \\ &= \max_\pi \mathbb{E}[R(s,a,s') + \gamma V^\pi(s') | s_t = s, a_t = a] \\ &= \mathbb{E}\left[R(s,a,s') + \gamma \max_\pi V^\pi(s') \, | s_t = s, a_t = a\right] \\ &= \mathbb{E}[R(s,a,s') + \gamma V^*(s') | s_t = s, a_t = a] \end{aligned}$$

$\because Q^\pi(s,a) = \mathbb{E}[R(s,a,s') + \gamma V^\pi(s') | s_t = s, a_t = a]$

Summary

State-value function & State-action value function allows
Optimum Planning as a Greedy Search!

- **Reconstructing optimal policy with $Q^*(s,a)$ and $V^*(s)$**

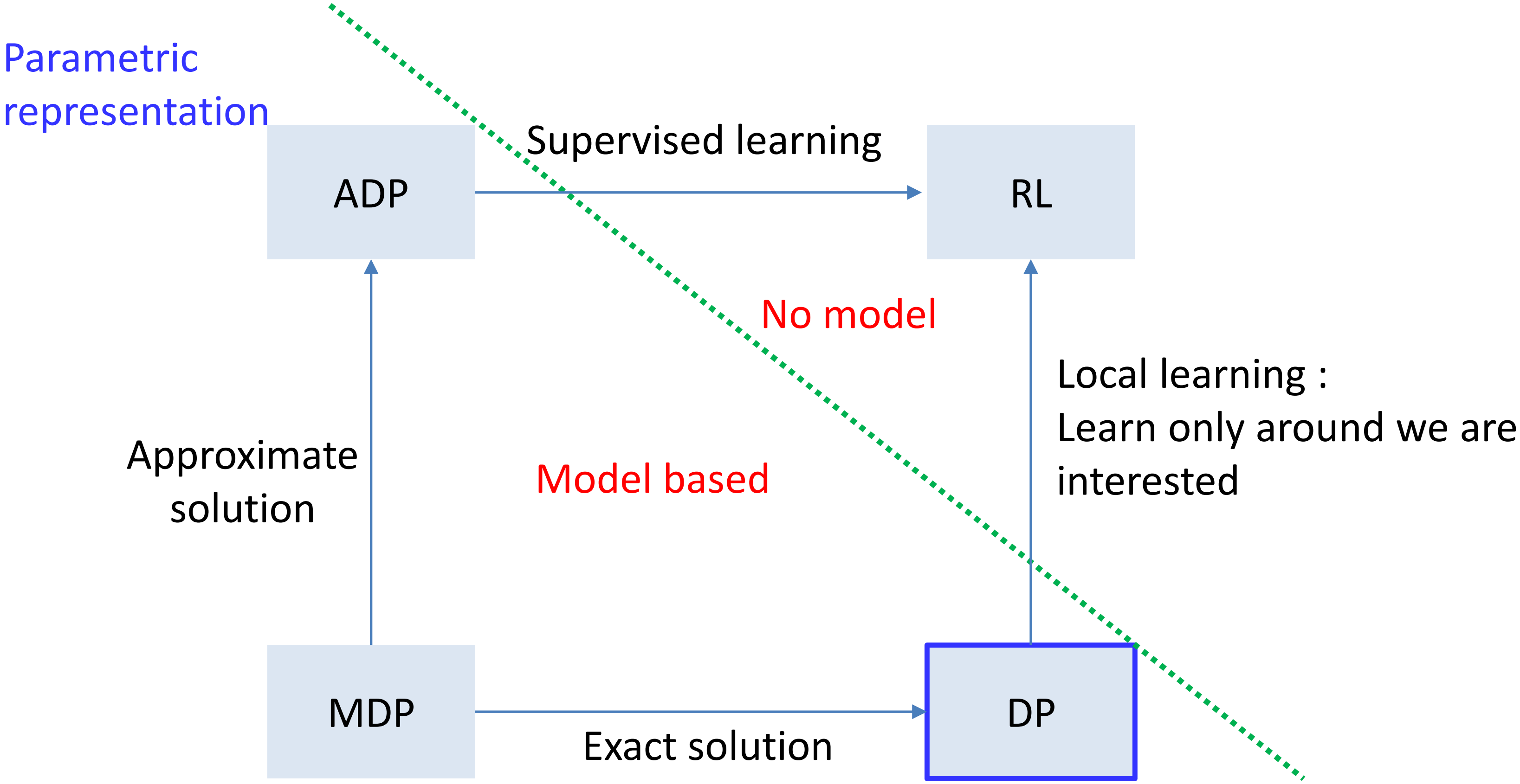
$$a^* = \operatorname*{argmax}_a Q^*(s,a)$$
$$= \operatorname*{argmax}_a \mathbb{E}[R(s,a,s') + \gamma V^*(s') | s_t = s, a_t = a]$$

- Any greedy policy with respect to the optimal value function $V^*(s)$ is an optimal policy

  → because $V^*(s)$ already takes into account the reward consequences of all possible future behavior

- The Q function effectively catches the results of all one-step-ahead search

Road Map for Next Lectures



- Usually Tabular representation
- For special case, DP can be used for solving a MDP with continuous space and action

- All other methods can be viewed as attempts to achieve much the same effect as DP, only with less computation and without assuming a perfect model of the environment