# Markov decision processes and foundations of reinforcement learning

# From last lecture (A-C model)

$$\frac{1}{\tau^2}\Big(x_{j-1} - 2x_j + x_{j+1}\Big) \;=\; \tilde{\kappa}^2\, x_j, \tag{16}$$

with

$$\tilde{\kappa}^2 \;=\; \frac{\lambda\sigma^2}{\tilde{\eta}} \;=\; \frac{\lambda\sigma^2}{\eta\left(1 - \dfrac{\gamma\tau}{2\eta}\right)}.$$

Note that equation (16) is a linear difference equation whose solution may be written as a combination of the exponentials $\exp(\pm\kappa t_j)$, where $\kappa$ satisfies

$$\frac{2}{\tau^2}\Big(\cosh(\kappa\tau) - 1\Big) \;=\; \tilde{\kappa}^2.$$

The tildes on $\tilde{\eta}$ and $\tilde{\kappa}$ denote an $\mathcal{O}(\tau)$ correction; as $\tau \to 0$ we have $\tilde{\eta} \to \eta$ and $\tilde{\kappa} \to \kappa$. The specific solution with $x_0 = X$ and $x_N = 0$ is a trading trajectory of the form:

$$x_j \;=\; \frac{\sinh\big(\kappa(T - t_j)\big)}{\sinh\big(\kappa T\big)}\, X, \qquad\qquad j = 0, \ldots, N, \tag{17}$$

and the associated trade list

$$n_j \;=\; \frac{2\sinh\big(\frac{1}{2}\kappa\tau\big)}{\sinh\big(\kappa T\big)}\, \cosh\Big(\kappa\Big(T - t_{j-\frac{1}{2}}\Big)\Big)\, X, \qquad j = 1, \ldots, N, \tag{18}$$

where sinh and cosh are the hyperbolic sine and cosine functions, and

# Processes and States

- **Process**: time-sequenced random outcome
- Random outcome example: price of a derivative, portfolio value etc
- **State**: Internal Representation $S_t$ driving future evolution
- We are interested in $P[S_{t+1} | S_t, S_{t-1}, \ldots, S_0]$

# Markov Property

- The future is independent of the past given the present
- $P[X_{t+1}| X_t ,X_{t-1}, \dots ,X_0] = P[X_{t+1}|X_t ]$ for all $t>0$

- This makes the mathematics easier and the computation tractable
- We call this the **Markov Property of States**
- The state captures all relevant information from history
- Once the state is known, the history may be thrown away

# Markov Processes (or Markov Chains)

**Definition**

A *Markov Process* consists of:

- A countable set of states $\mathcal{S}$ (known as the State Space) and a set $\mathcal{T} \subseteq \mathcal{S}$ (known as the set of Terminal States)
- A time-indexed sequence of random states $S_t \in \mathcal{S}$ for time steps $t = 0, 1, 2, \ldots$ with each state transition satisfying the Markov Property: $\mathbb{P}[S_{t+1}|S_t, S_{t-1}, \ldots, S_0] = \mathbb{P}[S_{t+1}|S_t]$ for all $t \geq 0$
- Termination: If an outcome for $S_T$ (for some time step $T$) is a state in the set $\mathcal{T}$, then this sequence outcome terminates at time step $T$

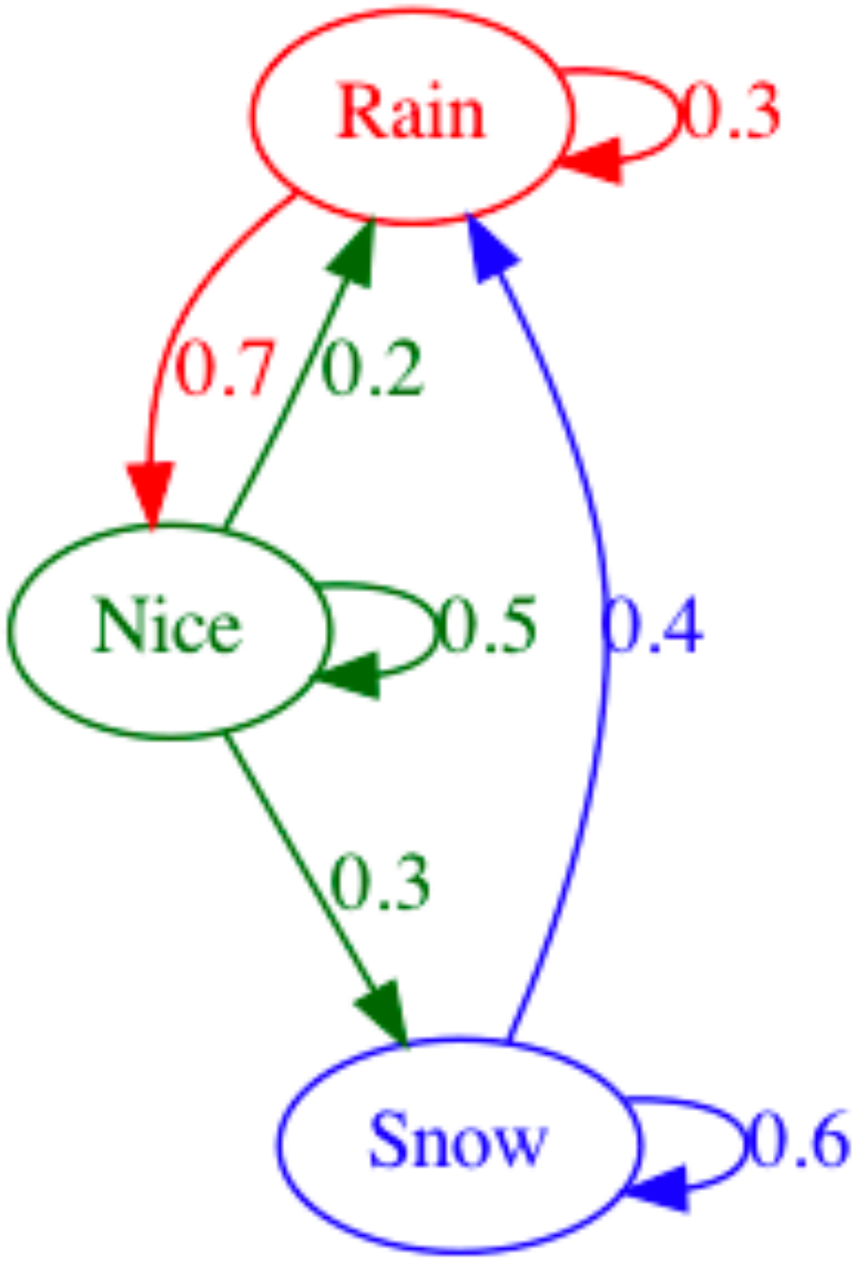
- The more commonly used term for *Markov Process* is *Markov Chain*
- We refer to $\mathbb{P}[S_{t+1}|S_t]$ as the transition probabilities for time $t$.
- Non-terminal states: $\mathcal{N} = \mathcal{S} - \mathcal{T}$

# Transition Probability Matrix

$$p_{ij} = P(X_{t+1} = j \,|\, X_t = i)$$

$$P = \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n1} & p_{n2} & \cdots & p_{nn} \end{pmatrix}$$

# Example: Weather Finite Markov Process

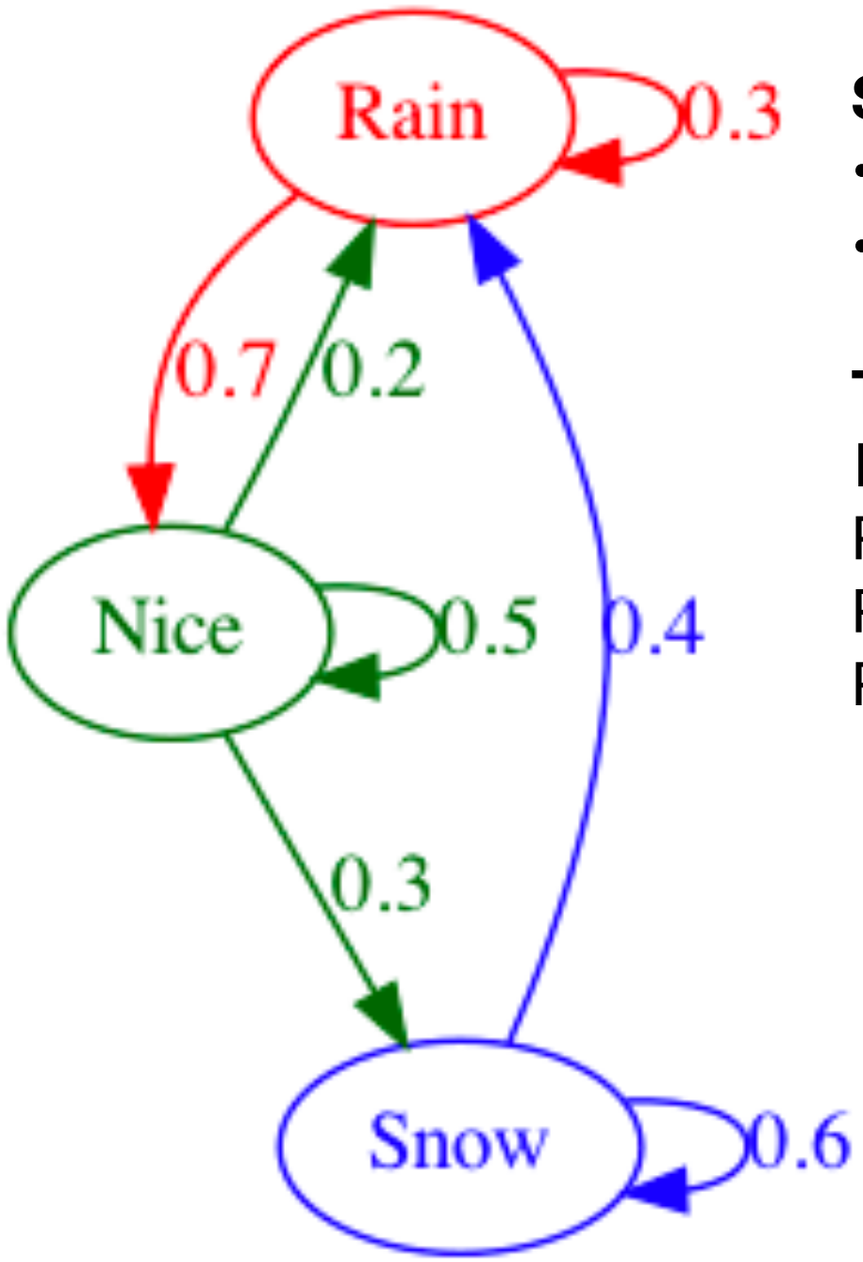



**States =**

- **Terminal States =**
- **Non-Terminal States =**

**Transition Probabilities:**

# Example: Weather Markov Process




**States** = {Rain, Nice, Snow}

- **Terminal States =** none
- **Non-Terminal States =** {Rain, Nice, Snow}

**Transition Probabilities:**

$P_{Nice,Rain} = 0.2$

$P_{Nice,Nice} = 0.5$

$P_{Nice,Snow} = 0.3$

$P_{Nice,Rain} + P_{Nice,Nice} + P_{Nice,Snow} = 1$

# Example: Snakes and Ladders Game (single player)

- https://toytheater.com/snakes-and-ladders/

**States =**

- **Terminal States =**
- **Non-Terminal States =**

**Transition Probabilities:**

**Play ALONE**
**Ladder - go up, snake - go down**
**Win when cross 100**
**Rolling 6 gives a player an extra die roll**

# Example: Snakes and Ladders Game (single player)

- https://toytheater.com/snakes-and-ladders/

**Play ALONE**
**Ladder - go up, snake - go down**
**Win when cross 100**
**Rolling 6 gives a player an extra die roll**

**States** = {1, 2 ,3 ,4 ,5 ,…., 100}

- **Terminal States** = {100}
- **Non-Terminal States** = {1, … , 99}

**Transition Probabilities:**

$P_{1,1} = 0$

$P_{1,2}=P_{1,3}=P_{1,5}=P_{1,6}= P_{1,7}= 1/6$

$P_{1,14} = 1/6$

$P_{1,4} = 0$

$P_{1,8}= \ldots = P_{1,100} = 0$

$P_{97,98} = 0$

$P_{97,78} = 1/6$

$P_{97,99} = 1/6$

$P_{97,100} = 2/3$

# Example: Mouse and Cheese

- 🧀 +1000 points
- 💧 +10 points
- ⚡ -100 points

- **<u>Goal:</u> Find best Mouse's policy to maximize its rewards**

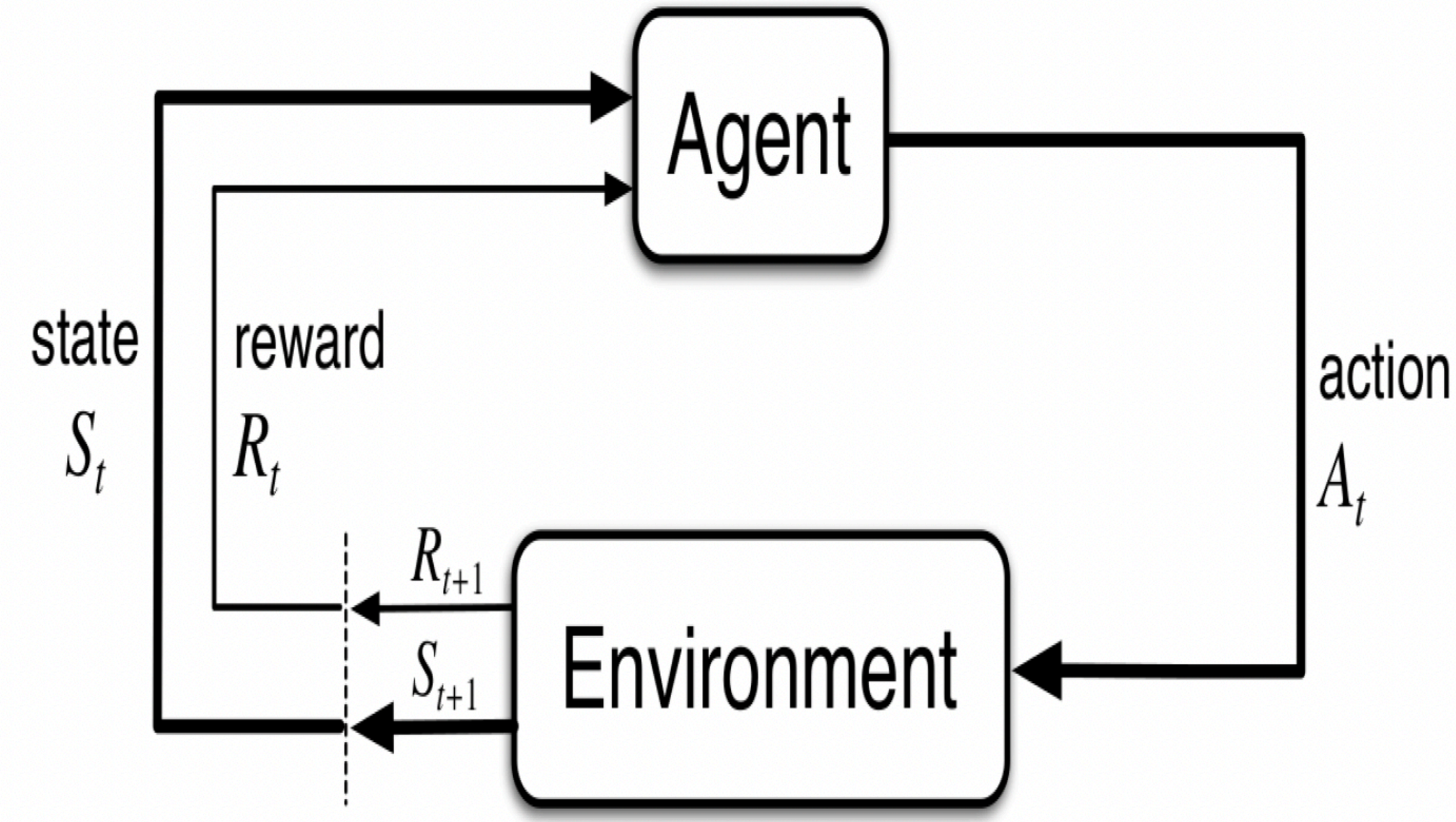
# Example: Mouse and Cheese

- 🧀 +1000 points
- 💧 +10 points
- ⚡ -100 points

- **<u>Goal:</u> Find best Mouse's policy to maximize its rewards**

- **<u>States:</u> {Positions of Mouse, cheese, water, lighting within the maze}**
- **<u>Actions</u>: {forward, back} in a corridor and {forward, back, left, right} at a crossroads**
- **<u>Transition between states</u>: implied by maze (deterministic or stochastic)**
- **<u>Rewards at a state for taking action</u>**

# Agent Environment Interface



# Markov Decision Process (MDP)

**Definition**

A *Markov Decision Process (MDP)* comprises of:

- A countable set of states $\mathcal{S}$ (State Space), a set $\mathcal{T} \subseteq \mathcal{S}$ (known as the set of Terminal States), and a countable set of actions $\mathcal{A}$
- A time-indexed sequence of *environment-generated* pairs of random states $S_t \in \mathcal{S}$ and random rewards $R_t \in \mathcal{D}$ (a countable subset of $\mathbb{R}$), alternating with *agent-controllable* actions $A_t \in \mathcal{A}$ for time steps $t = 0, 1, 2, \ldots$
- Markov Property: $\mathbb{P}[(R_{t+1}, S_{t+1})|(S_t, A_t, S_{t-1}, A_{t-1}, \ldots, S_0, A_0)] = \mathbb{P}[(R_{t+1}, S_{t+1})|(S_t, A_t)]$ for all $t \geq 0$
- Termination: If an outcome for $S_T$ (for some time step $T$) is a state in the set $\mathcal{T}$, then this sequence outcome terminates at time step $T$.

$$S_0, A_0, R_1, S_1, A_1, R_2, S_2, A_2, \ldots, S_{T-1}, A_{T-1}, R_T, S_T$$

# Markov Decision Process (MDP)

- Environment can be deterministic or stochastic (typically)
- Transition probabilities $P(S_{t+1} \mid S_t, a_t)$

# Rewards

- At state $s_t$, the agent takes an action $a_t$ to transfer to state $s_{t+1}$ and receives a numerical reward $R_{t+1}$ for it

- The agent is interested in maximizing total cumulative rewards
- Define the *Return* $G_t$ from state $S_t$ as:

$$G_t = \sum_{i=t+1}^{\infty} \gamma^{i-t-1} \cdot R_i = R_{t+1} + \gamma \cdot R_{t+2} + \gamma^2 \cdot R_{t+3} + \dots$$

- $\gamma \in [0, 1]$ is the discount factor. Why discount?
  - Mathematically convenient to discount rewards
  - Avoids infinite returns in cyclic Markov Processes
  - Uncertainty about the future may not be fully represented
  - If reward is financial, discounting due to interest rates
  - Animal/human behavior prefers immediate reward
- If all sequences terminate (Episodic Processes), we can set $\gamma = 1$

# Reinforcement learning (RL)

- At each time step, the agent implements a mapping from states to probabilities of selecting each possible action - called agents policy $\pi(s_t, a_t)$

- **Reinforcement learning (RL)** methods specify how the agent changes its policy as a result of experience

- Agent's goal: maximize total cumulative rewards - RL helps achieve it!

# Value Functions

- State-value function (value of state *s* under a policy $\pi$)

$$V^{\pi}(s) = E_{\pi}(R_t \,|\, s_t = s) = E_{\pi}(\sum_{k=0}^{\inf} \gamma^k r_{t+k+1} \,|\, s_t = s)$$

- Action-value function (value of taking action *a* in state *s* under a policy $\pi$)

$$Q^{\pi}(s, a) = E_{\pi}(R_t \,|\, s_t = s, a_t = a) = E_{\pi}(\sum_{k=0}^{\inf} \gamma^k r_{t+k+1} \,|\, s_t = s, a_t = a)$$

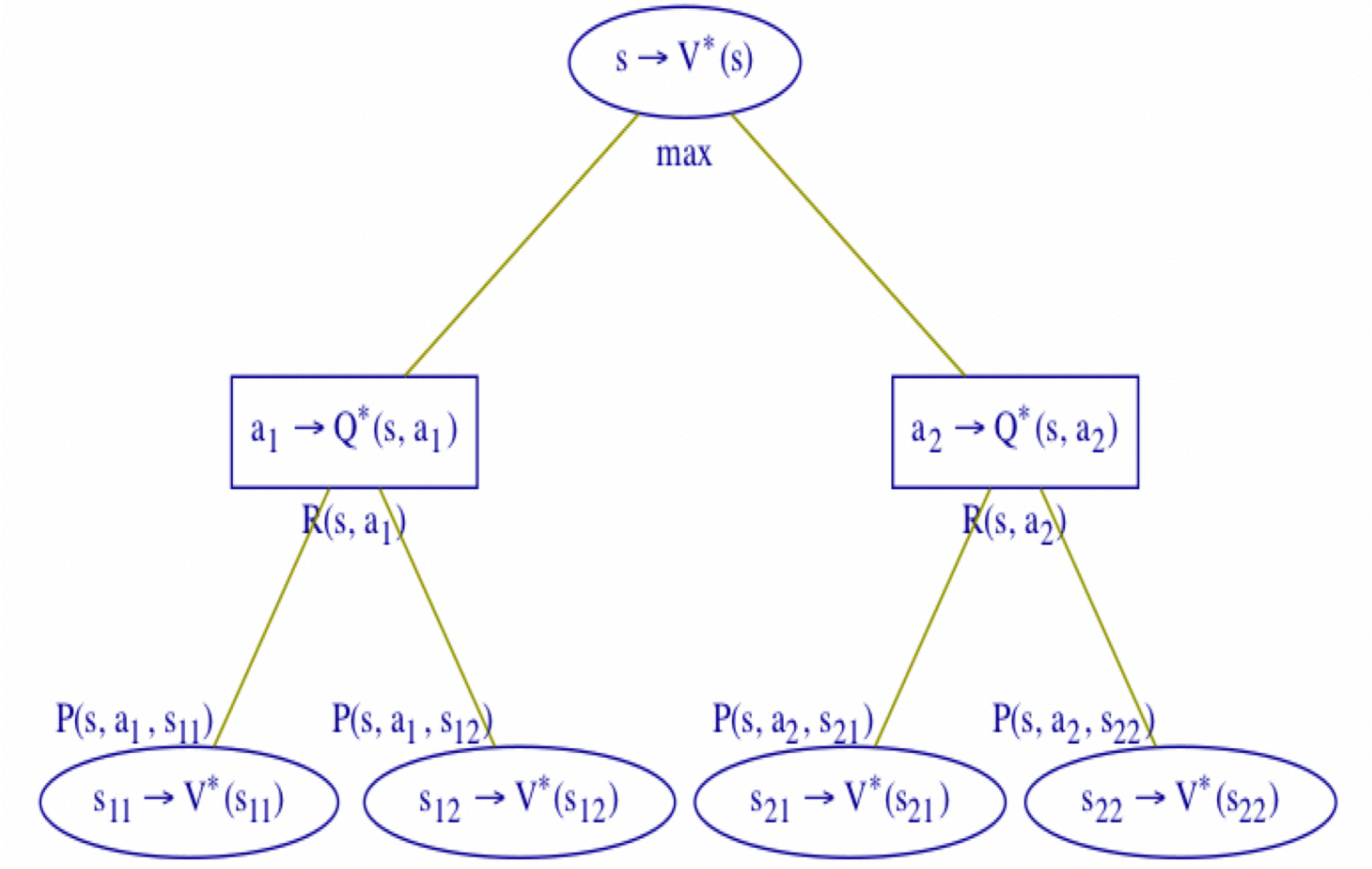
# Optimal Value Functions

- Interested in finding an optimal policy $\pi^*$
- Optimal state-value function $V^*(s) = \max_\pi V^\pi(s)$
- Optimal action-value function $Q^*(s,a) = \max_\pi Q^\pi(s,a)$

# Bellman Optimality Equation

$$
\begin{aligned}
V^*(s) = & \max_a Q^{\pi^*}(s,a) = \max_a E_{\pi^*}(R_t \,|\, s_t = s, a_t = a) \\
= & \max_a E_{\pi^*}(\sum_{k=0}^{\inf} \gamma^k r_{t+k+1} \,|\, s_t = s, a_t = a) \\
= & \max_a E_{\pi^*}(r_{t+1} + \sum_{k=0}^{\inf} \gamma^k r_{t+k+2} \,|\, s_t = s, a_t = a) \\
= & \max_a E_{\pi}(r_{t+1} + \sum_{k=0}^{\inf} \gamma^k r_{t+k+2} \,|\, s_t = s, a_t = a) \\
= & \max_a E_{\pi}(r_{t+1} + \gamma V^*(s_{t+1}) \,|\, s_t = s, a_t = a) \\
= & \max_a \sum_{s'} P^a_{ss'}(R^a_{ss'} + \gamma V^*(s'))
\end{aligned}
$$

# Bellman Equation (visualization)



# Model free vs. model based RL

- Bellman equation (state-action presentation):

$$
\begin{aligned}
Q^*(s,a) = \quad & E(r_{t+1} + \gamma \max_{a'} Q^*(s_{t+1}, a') \,|\, s_t = s, a_t = a) \\
= \quad & \sum_{s'} P^a_{ss'}(R^a_{ss'} + \gamma \max_{a'} Q^*(s', a'))
\end{aligned}
$$

- Explicit knowledge of transition probabilities not needed to learn the optimal policy

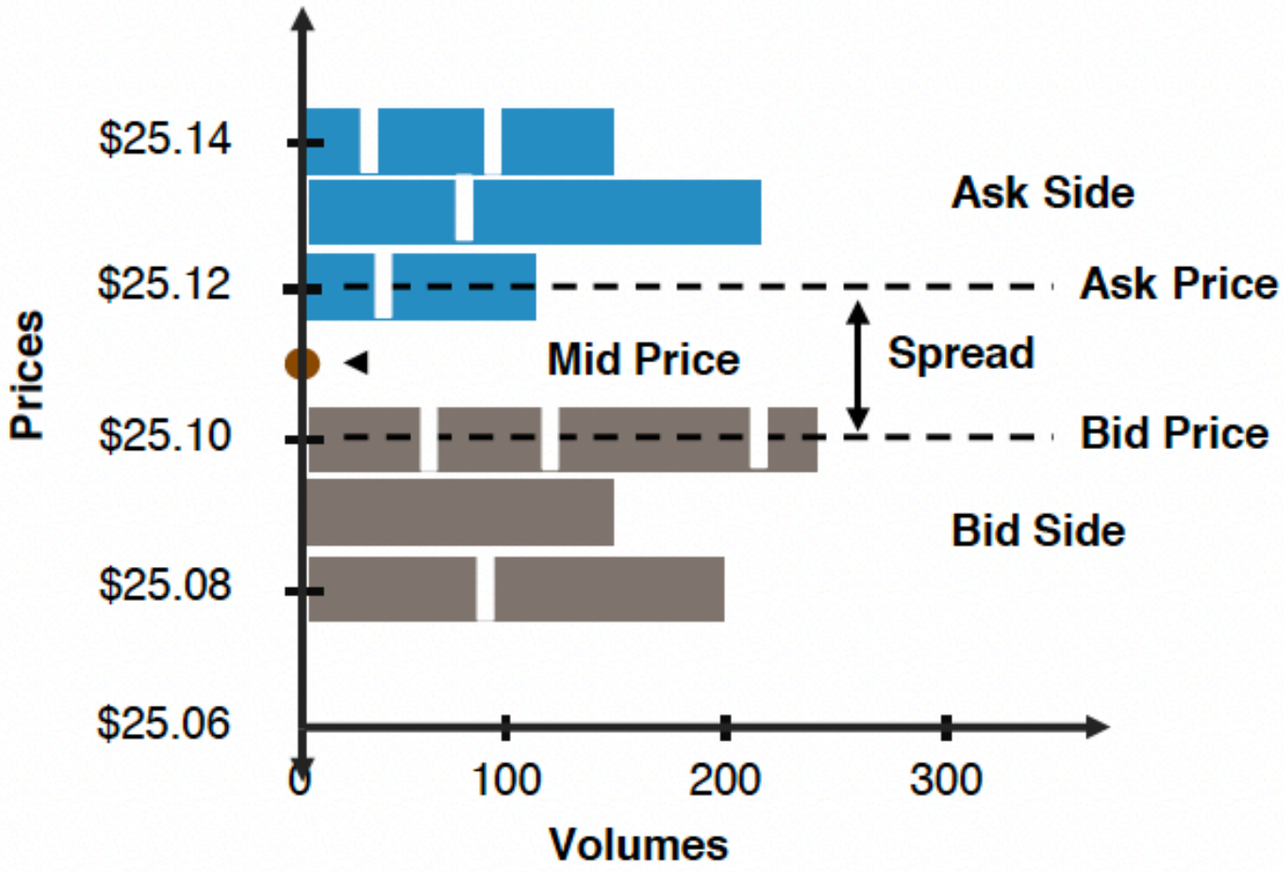
- Model free vs. model based RL

- Model free RL - learn in environment or the simulator!

# Exploration/Exploitation

- RL enables to learn by experience
- Explore in a simulator!
- Exploration is important for market problems

# MDPs in Limit Order Books




- **Execution vs. market maker**

- **States: ?**
- **Actions: ?**
- **Transition probabilities: ?**
- **Rewards: ?**

# Common Order Book States

- Imbalance $\frac{\textbf{bid volume}}{\textbf{ask volume}}$
- Volatilities
- Price moves at different time scales
- Spreads
- Traded volumes

# Common Order Book States

| Variable | Description |
|---|---|
| $P_t^{b,i}$ | the $i^{th}$ best log bid price just after the $t^{th}$ event |
| $P_t^{a,i}$ | the $i^{th}$ best log ask price just after the $t^{th}$ event |
| $G_t^{b,i}$ | the $i^{th}$ bid gap price just after the $t^{th}$ event |
| $S_t$ | the spread just after the $t^{th}$ event |
| $G_t^{a,i}$ | the $i^{th}$ ask gap price just after the $t^{th}$ event |
| $V_t^{b,i}$ | log volume of the $i^{th}$ best bid quote just after the $t^{th}$ event |
| $V_t^{a,i}$ | log volume of the $i^{th}$ best ask quote just after the $t^{th}$ event |
| $BLO_t$ | dummy variable equal to 1 if the $t^{th}$ event is a limit order event at bid side |
| $ALO_t$ | dummy variable equal to 1 if the $t^{th}$ event is a limit order event at ask side |
| $BMO_t$ | dummy variable equal to 1 if the $t^{th}$ event is a market order event at bid side |
| $AMO_t$ | dummy variable equal to 1 if the $t^{th}$ event is a market order event at ask side |
| $BTT_t$ | dummy variable equal to 1 if the $t^{th}$ event is a trade-through event at bid side |
| $ATT_t$ | dummy variable equal to 1 if the $t^{th}$ event is a trade-through event at ask side |

# Price jump prediction in Limit Order Book

Ban Zheng[*], Eric Moulines[**], Frédéric Abergel[***]