# Reinforcement Learning

- MDP & Bellman equation

  - Markov Decision Process

    - MDP is a model for sequential stochastic decision problems
    - with four tuples $<S, A, R, tr>$ (State, Action, Reward, Transition probability)

  - Bellman Equation

$$v_\pi(s) \doteq \sum_{a \in A} \pi(a|s) \sum_{s',r} p(s', r|s, a)[r + \gamma v_\pi(s')], \quad \text{for all} \quad s \in S$$

    - state - value function

$$v_\pi(s) \doteq \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \,\middle|\, S_t = s\right], \text{ for all } s \in \mathcal{S},$$

    - action - value function

$$q_\pi(s, a) \doteq \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \,\middle|\, S_t = s, A_t = a\right]$$

# Reinforcement Learning

- Q - Learning

  - Off-Policy Temporal-Difference Control

    - differentiate behavior-policy from learning-policy

    - SARSA (on policy)
      - *<s, a, r, s', a' ← π(s)>* => Learning

    - Q-Learning
      - *<s, a ← π(s), r, s'>* => Learning

    - Update rule

$$Q(S_t, A_t)_{new} = Q(S_t, A_t)_{old} + \alpha[R_{t+1} + \gamma max_{a \in A} Q(S_{t+1}, a') - Q(S_t, A_t)_{old}]$$