REINFORCEMENT LEARNING

Introduction to Reinforcement Learning and Markov Decision Process

-Vishal Kumar
dreamerkumar.com

Learning to Bike

-Vishal Kumar
dreamerkumar.com

KEY INSIGHTS

- Learning to achieve goals by interacting with the environment
- Inspired by biological learning systems
- Closest to the kind of learning that humans and other animals do
- Eg: Emergence of Locomotion Behaviors in Rich Environments

-Vishal Kumar
dreamerkumar.com

COMPUTATIONAL APPROACH

- Map situations to actions
- Take actions to maximize a numerical reward
  - Analogous to experiences of pleasure or pain in biological systems
- Maximize the total reward over the long run

-Vishal Kumar
dreamerkumar.com

DIFFERENT FROM SUPERVISED LEARNING

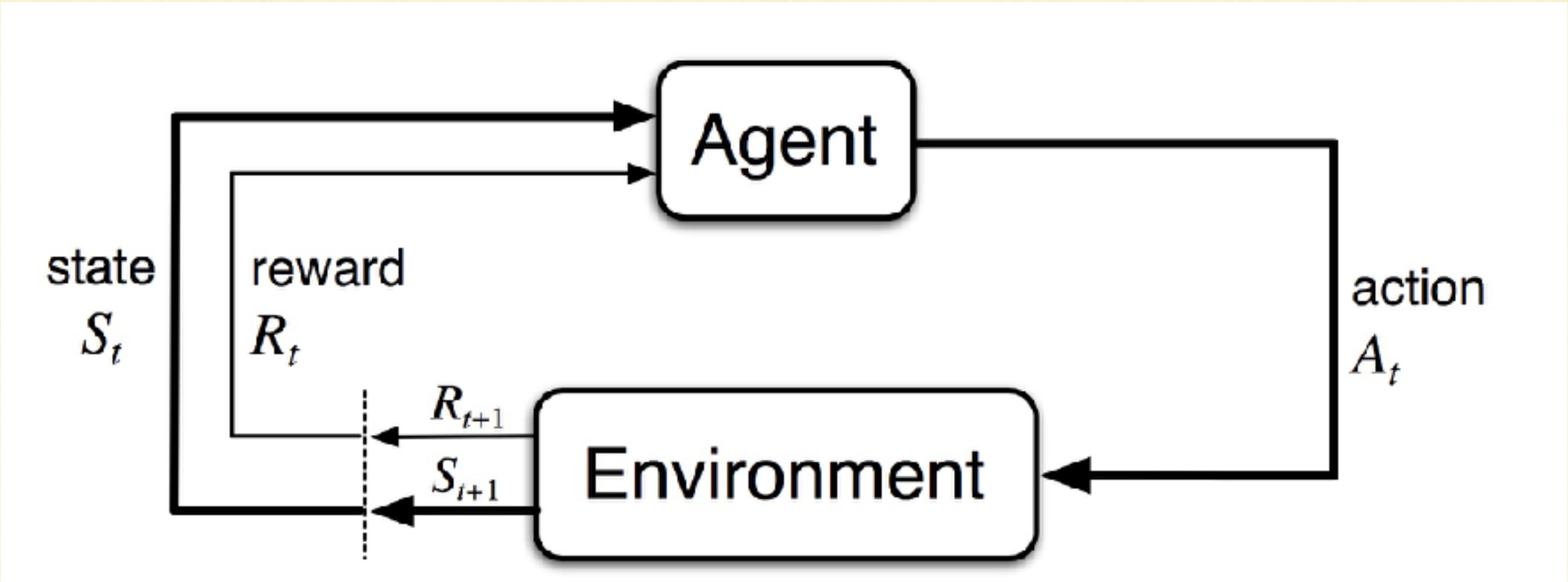
- In SL
  - The system generalizes the responses to act correctly in situations not present in the training set
  - Has examples of desired behavior that are both correct and representative of all the situations
- RL
  - Agent is in unchartered territory where it must be able to learn from it's own experience

DIFFERENT FROM UNSUPERVISED LEARNING

- In UL
  - We find structure hidden in collections of unlabeled data
- In RL
  - We try to maximize a reward signal

-Vishal Kumar
dreamerkumar.com

AGENT ENVIRONMENT INTERACTION IN RL



-Vishal Kumar
dreamerkumar.com

GOAL OF RL ALGORITHMS

- Find the optimal policy:
  - The best action to take at each of the states that the agent ends up in
  - This is determined by taking action that gives the maximum total reward

-Vishal Kumar
dreamerkumar.com

CALCULATING TOTAL REWARDS

$$G_t \doteq R_{t+1} + R_{t+2} + R_{t+3} + \cdots + R_T$$

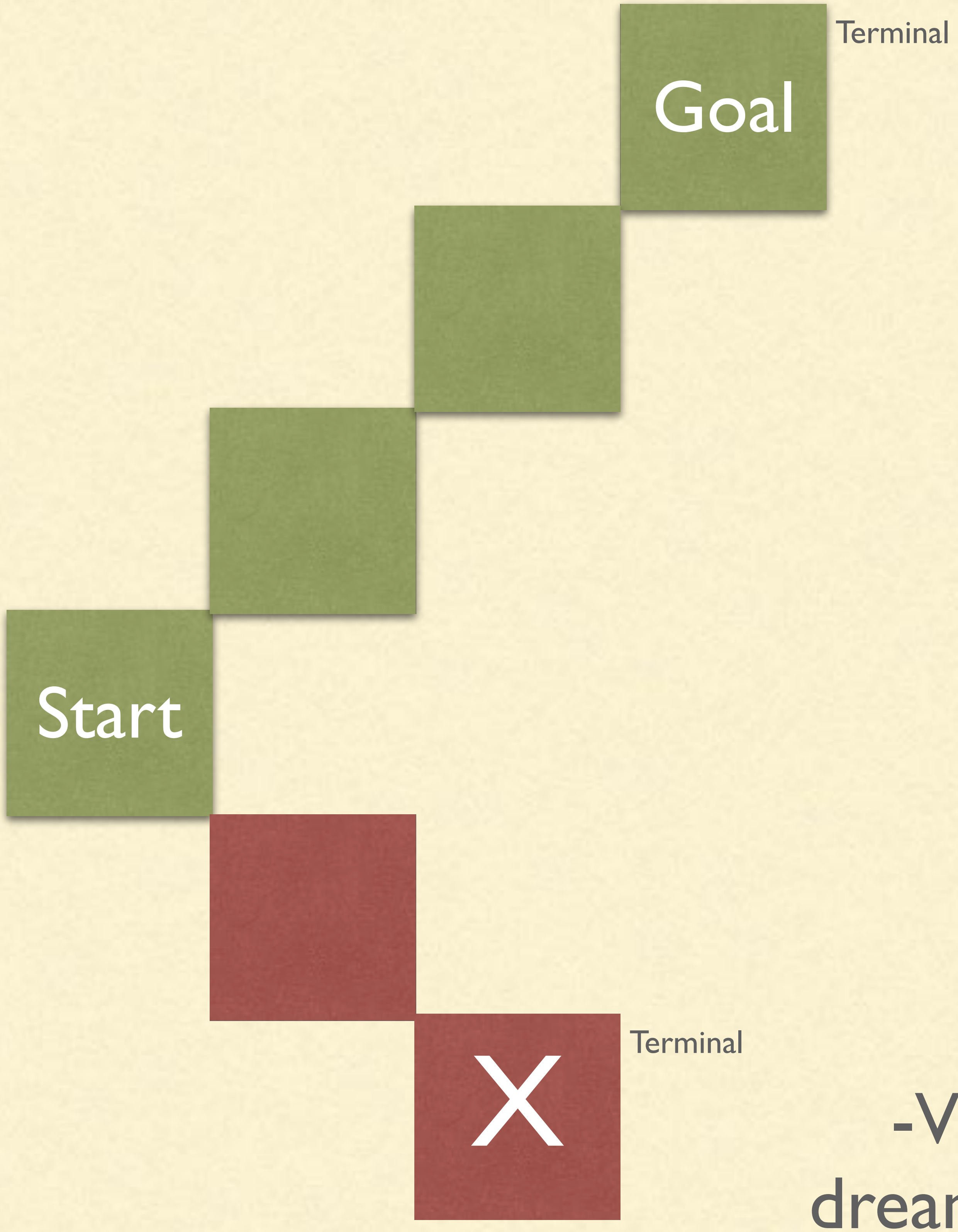
DISCOUNTED SUM OF REWARDS

$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots \quad = \quad \sum_{k=0}^{\infty} \gamma^k R_{t+k+1},$$

where γ is a parameter, $0 \leq \gamma \leq 1$, called the *discount rate*.

-Vishal Kumar
dreamerkumar.com

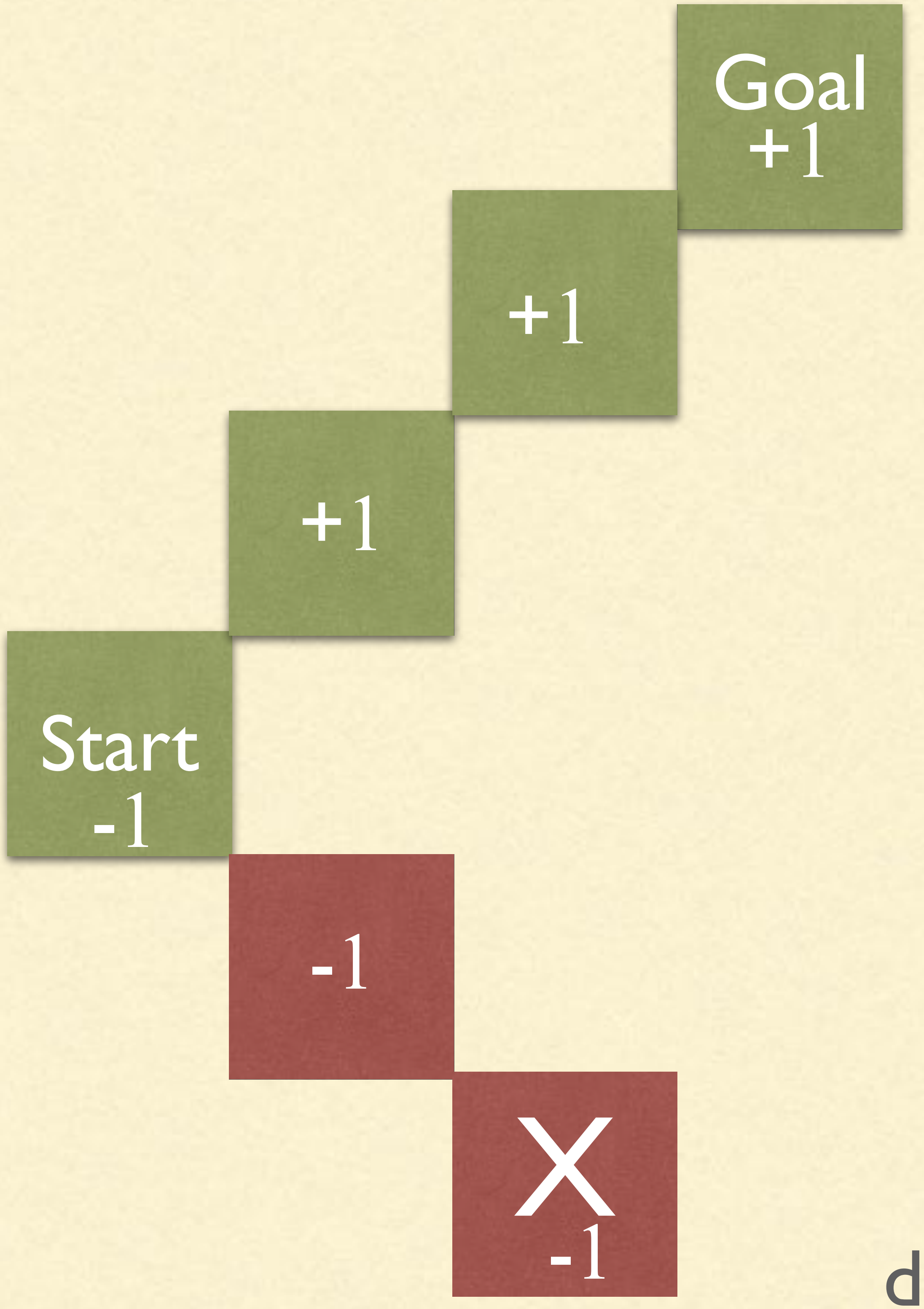
LETS PLAY WITH REWARDS TO GET OPTIMAL POLICY



-Vishal Kumar

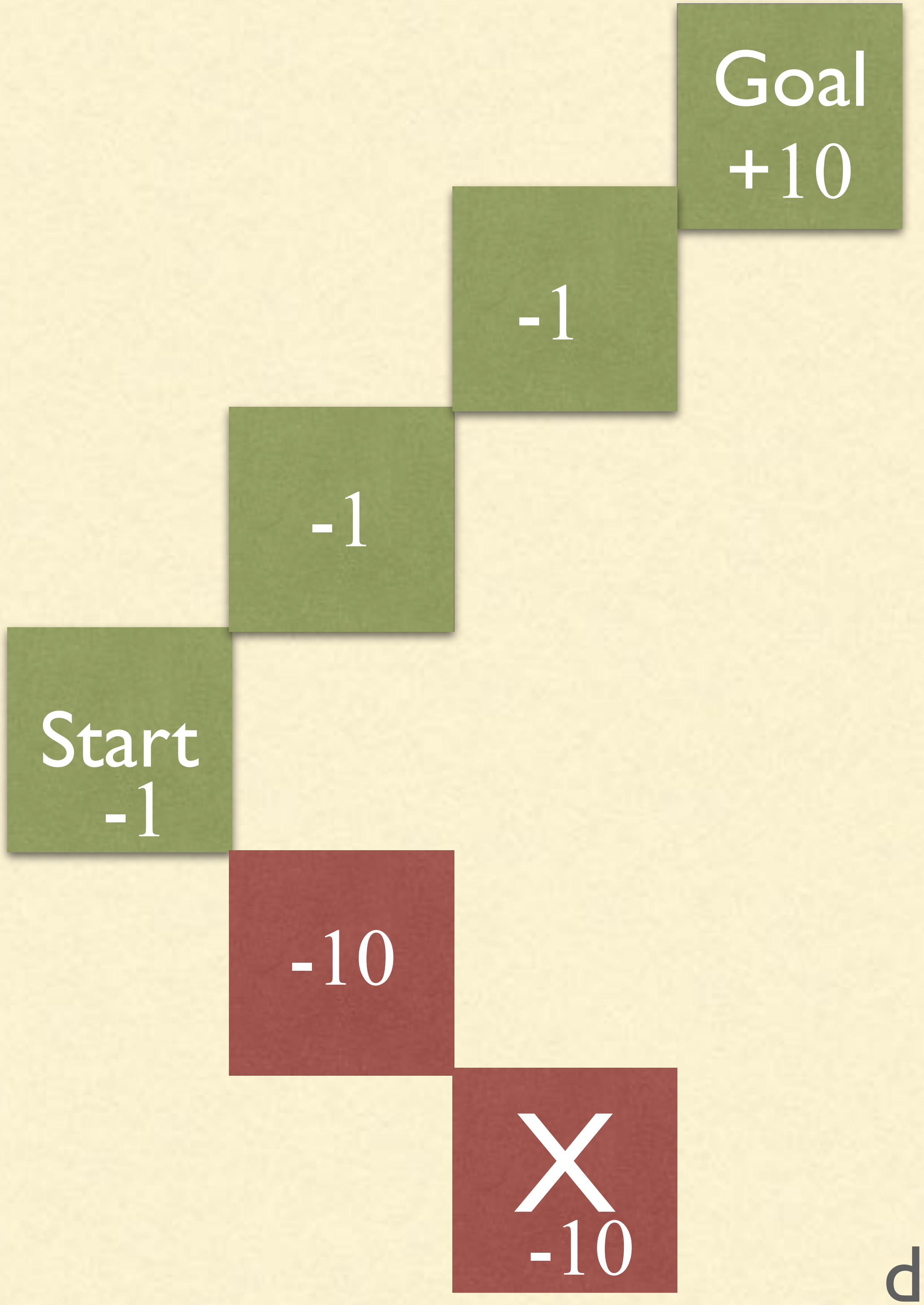
dreamerkumar.com

INCORRECT REWARD ASSIGNMENT



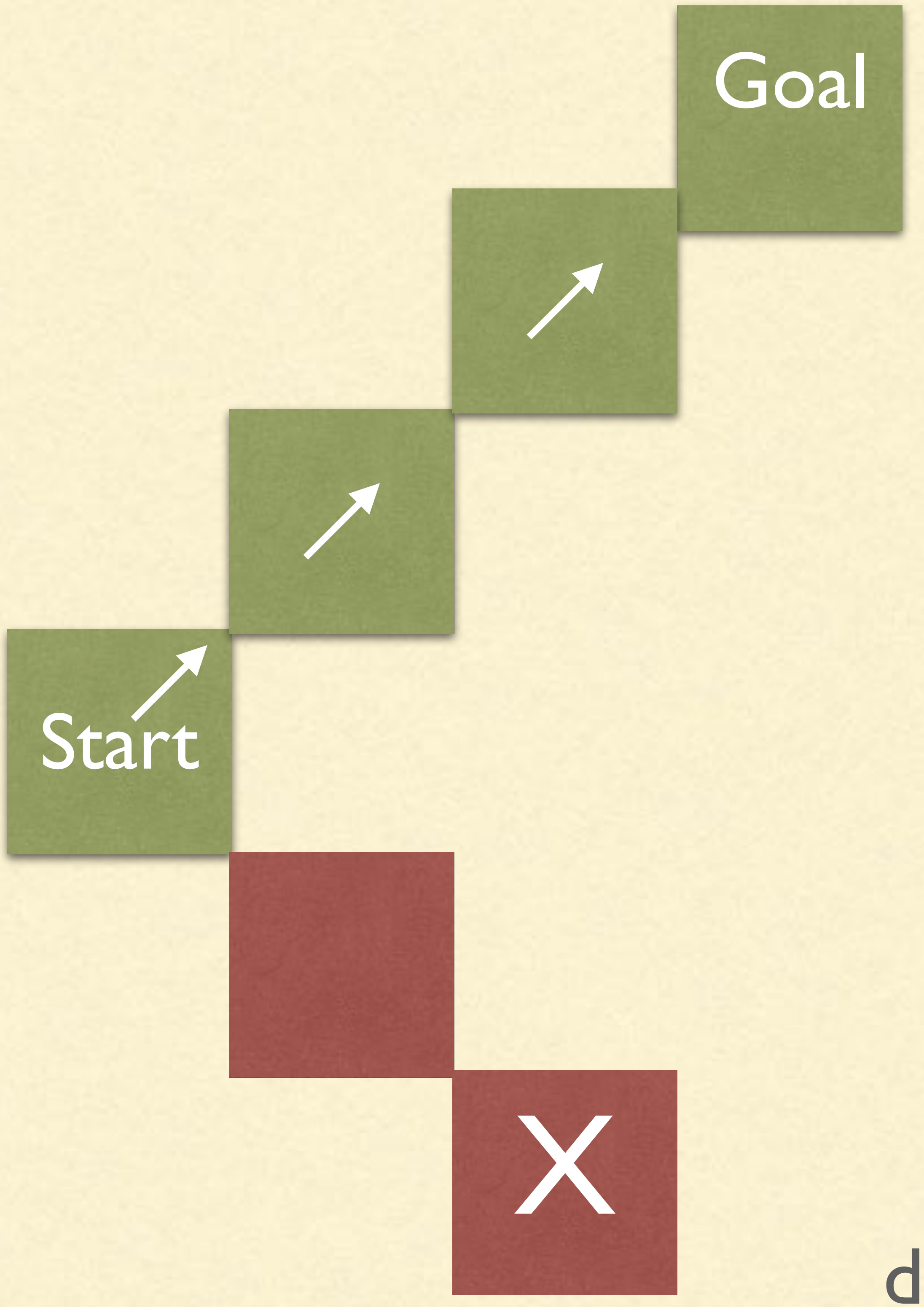
-Vishal Kumar
dreamerkumar.com

CORRECT REWARD ASSIGNMENT



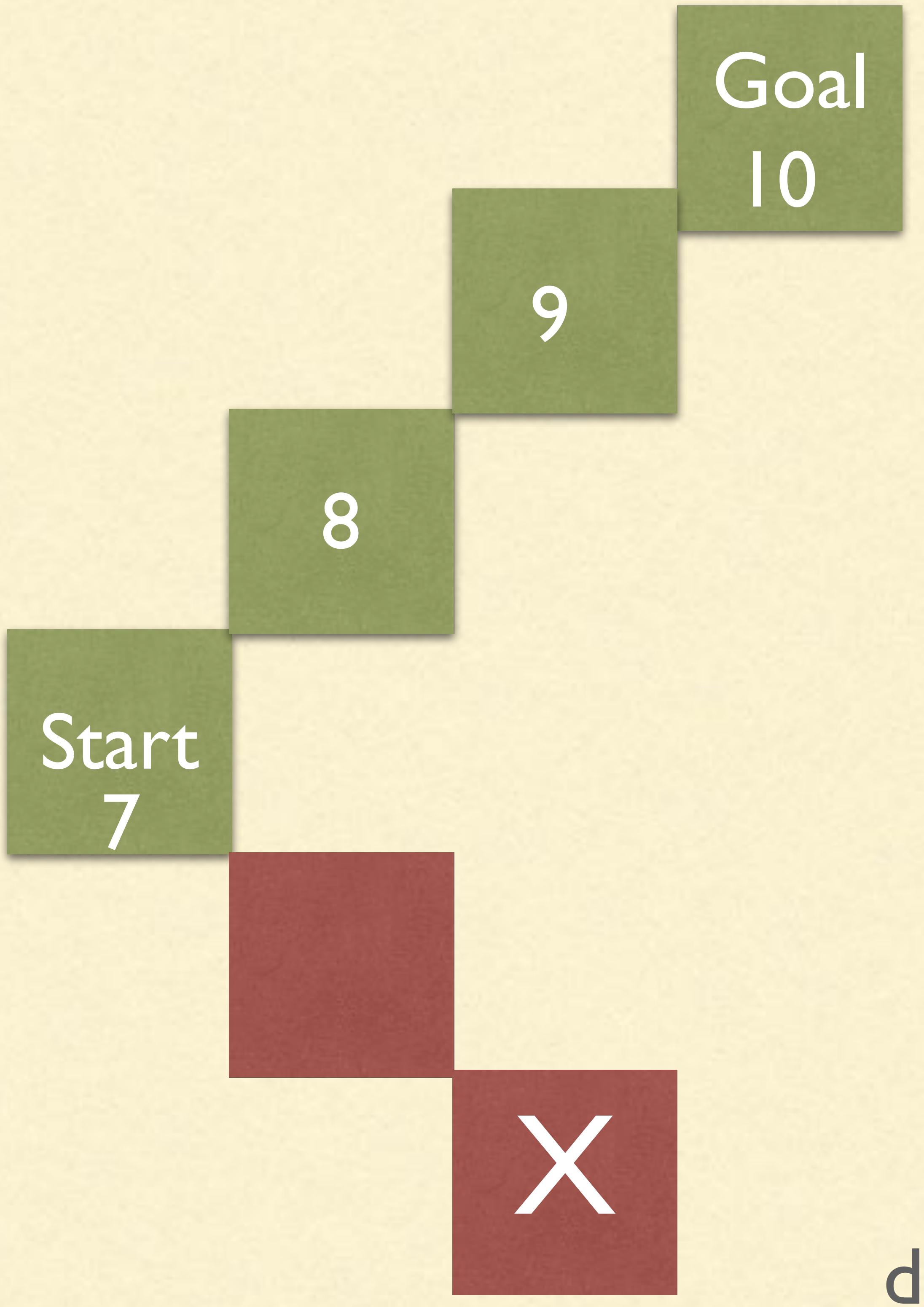
-Vishal Kumar
dreamerkumar.com

OPTIMAL POLICY



-Vishal Kumar
dreamerkumar.com

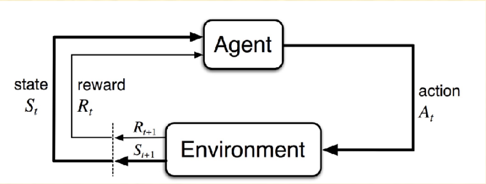
STATE VALUES FOR OPTIMAL POLICY



-Vishal Kumar
dreamerkumar.com

ELEMENTS OF REINFORCEMENT LEARNING

- Policy
- Reward Signal
- Value Function
- Action Value Function
- Model of the environment



-Vishal Kumar
dreamerkumar.com

VALUE FUNCTION = EXPECTED SUM OF REWARDS

$$G_t \doteq R_{t+1} + R_{t+2} + R_{t+3} + \cdots + R_T$$

DISCOUNTED SUM OF REWARDS

$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots \quad = \quad \sum_{k=0}^{\infty} \gamma^k R_{t+k+1},$$

where γ is a parameter, $0 \leq \gamma \leq 1$, called the *discount rate*.

-Vishal Kumar
dreamerkumar.com

VALUE FUNCTION FOR STOCHASTIC ENVIRONMENT

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

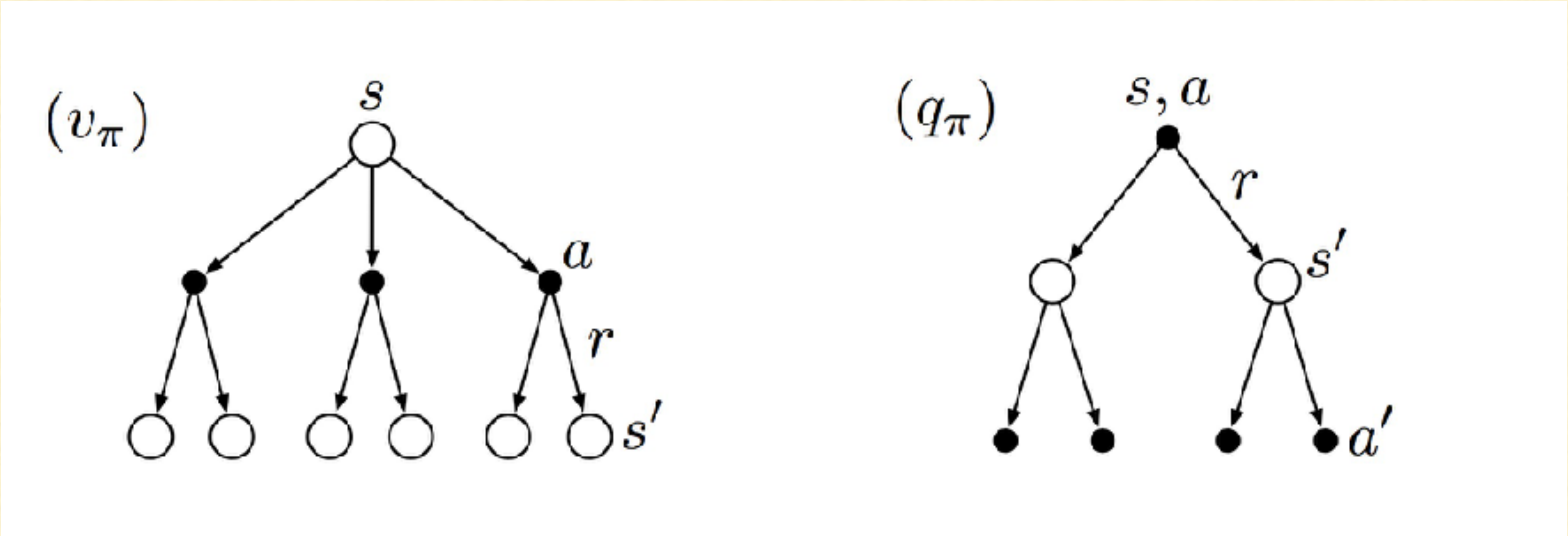
ACTION VALUE FUNCTION

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

-Vishal Kumar

dreamerkumar.com

BACKUP DIAGRAM FOR VALUE FUNCTION AND ACTION VALUE FUNCTION



-Vishal Kumar
dreamerkumar.com

FROZEN LAKE PROBLEM - SAVE THE FRISBY

Berkeley Deep RL Class [HW2](HW2) [(license)](license)

S	F	F	F
F	H	F	H
F	F	F	H
H	F	F	G

```
State transitions as array of (probability, next_state & reward)
_______________________________________________________________
P[0][0] = [(0.1, 0, 0.0), (0.8, 0, 0.0), (0.1, 4, 0.0)]
P[0][1] = [(0.1, 0, 0.0), (0.8, 4, 0.0), (0.1, 1, 0.0)]
P[0][2] = [(0.1, 4, 0.0), (0.8, 1, 0.0), (0.1, 0, 0.0)]
P[0][3] = [(0.1, 1, 0.0), (0.8, 0, 0.0), (0.1, 0, 0.0)]
P[1][0] = [(0.1, 1, 0.0), (0.8, 0, 0.0), (0.1, 5, 0.0)]
P[1][1] = [(0.1, 0, 0.0), (0.8, 5, 0.0), (0.1, 2, 0.0)]
P[1][2] = [(0.1, 5, 0.0), (0.8, 2, 0.0), (0.1, 1, 0.0)]
P[1][3] = [(0.1, 2, 0.0), (0.8, 1, 0.0), (0.1, 0, 0.0)]
P[2][0] = [(0.1, 2, 0.0), (0.8, 1, 0.0), (0.1, 6, 0.0)]
P[2][1] = [(0.1, 1, 0.0), (0.8, 6, 0.0), (0.1, 3, 0.0)]
P[2][2] = [(0.1, 6, 0.0), (0.8, 3, 0.0), (0.1, 2, 0.0)]
P[2][3] = [(0.1, 3, 0.0), (0.8, 2, 0.0), (0.1, 1, 0.0)]
P[3][0] = [(0.1, 3, 0.0), (0.8, 2, 0.0), (0.1, 7, 0.0)]
P[3][1] = [(0.1, 2, 0.0), (0.8, 7, 0.0), (0.1, 3, 0.0)]
P[3][2] = [(0.1, 7, 0.0), (0.8, 3, 0.0), (0.1, 3, 0.0)]
P[3][3] = [(0.1, 3, 0.0), (0.8, 3, 0.0), (0.1, 2, 0.0)]
P[4][0] = [(0.1, 0, 0.0), (0.8, 4, 0.0), (0.1, 8, 0.0)]
P[4][1] = [(0.1, 4, 0.0), (0.8, 8, 0.0), (0.1, 5, 0.0)]
P[4][2] = [(0.1, 8, 0.0), (0.8, 5, 0.0), (0.1, 0, 0.0)]
P[4][3] = [(0.1, 5, 0.0), (0.8, 0, 0.0), (0.1, 4, 0.0)]
P[5][0] = [(1.0, 5, 0)]
P[5][1] = [(1.0, 5, 0)]
P[5][2] = [(1.0, 5, 0)]
P[5][3] = [(1.0, 5, 0)]
P[6][0] = [(0.1, 2, 0.0), (0.8, 5, 0.0), (0.1, 10, 0.0)]
```

-Vishal Kumar
dreamerkumar.com

BELLMAN OPTIMALITY EQUATION

Bellman Equation

$$V^{\pi}(s) = R(s, \pi(s)) + \gamma \sum_{s'} P(s'|s, \pi(s)) V^{\pi}(s').$$

Bellman Optimality Equation

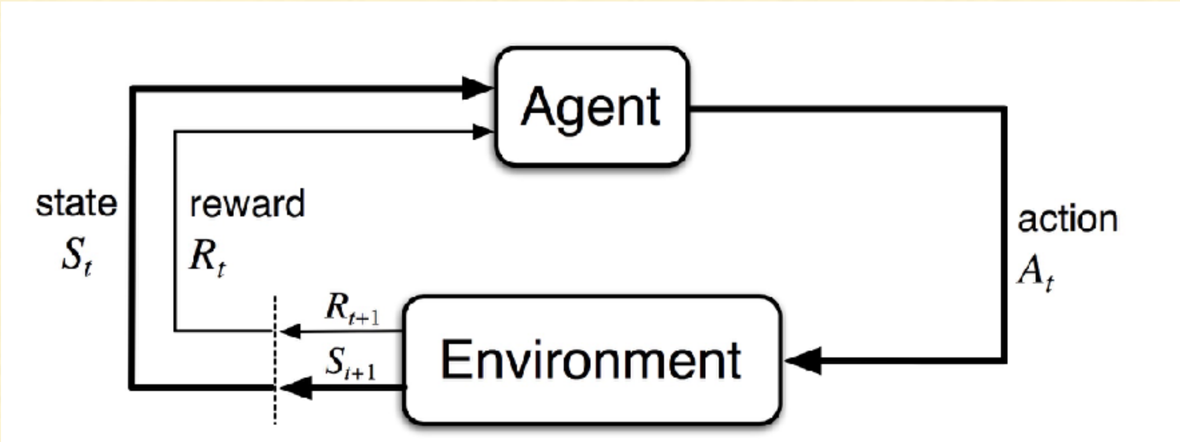
$$V^{*}(s) = \max_{a}\{R(s, a) + \gamma \sum_{s'} P(s'|s, a) V^{*}(s')\}.$$

MARKOV DECISION PROCESS

-Vishal Kumar
dreamerkumar.com

STATE

- State is a function of history $\Pr\{S_{t+1}=s', R_{t+1}=r \mid S_0, A_0, R_1, \ldots, S_{t-1}, A_{t-1}, R_t, S_t, A_t\}$
- Agent State is all Information available to the agent at a given time step t
  - It may not be all the information of the actual environment
  - It is only the information that the agent can extract through the interactions



-Vishal Kumar
dreamerkumar.com

MARKOV PROPERTY

- A state signal that succeeds in retaining all relevant information is said to be Markov
- Environment's response at **t+1** depends only on the state and action representations at **t**

$$\Pr\{S_{t+1} = s', R_{t+1} = r \mid S_0, A_0, R_1, \ldots, S_{t-1}, A_{t-1}, R_t, S_t, A_t\}$$

$$p(s', r|s, a) \doteq \Pr\{S_{t+1} = s', R_{t+1} = r \mid S_t = s, A_t = a\},$$

-Vishal Kumar
dreamerkumar.com

MARKOV DECISION PROCESS

Expected Reward for state-action pairs

$$r(s, a) \doteq \mathbb{E}[R_{t+1} \mid S_t = s, A_t = a] = \sum_{r \in \mathcal{R}} r \sum_{s' \in \mathcal{S}} p(s', r | s, a)$$

State Transition Probabilities

$$p(s'|s, a) \doteq \Pr\{S_{t+1} = s' \mid S_t = s, A_t = a\} = \sum_{r \in \mathcal{R}} p(s', r|s, a)$$

Expected Reward for state-action-next-state triples

$$r(s, a, s') \doteq \mathbb{E}\left[R_{t+1} \mid S_t = s, A_t = a, S_{t+1} = s'\right] = \frac{\sum_{r \in \mathcal{R}} r\, p(s', r|s, a)}{p(s'|s, a)}$$

-Vishal Kumar
dreamerkumar.com

FOR ANY MARKOV DECISION PROCESS

- There exists an optimal policy π_* that is better than or equal to all other policies,
    - $\pi_* \geq \pi, \forall \pi$
- All optimal policies achieve the optimal value function,
    - $v_{\pi_*}(s) = v_*(s)$
- All optimal policies achieve the optimal action-value function
    - $q_{\pi_*}(s, a) = q_*(s, a)$

-Vishal Kumar
dreamerkumar.com

OPTIMAL VALUE FUNCTIONS

OPTIMAL STATE VALUE FUNCTION

$$v_*(s) \doteq \max_{\pi} v_\pi(s)$$

OPTIMAL ACTION VALUE FUNCTION

$$q_*(s, a) \doteq \max_{\pi} q_\pi(s, a)$$

-Vishal Kumar
dreamerkumar.com

EXPLORATION/EXPLOITATION DILEMMA

- To receive data agent might have to take non optimal actions
  - Exploit Rewards currently available
  - But also explore states that could potentially give more rewards
- Stochastic (Random) Policies
  - Epsilon Greedy Algorithms

-Vishal Kumar
dreamerkumar.com

WHAT'S NEXT

- Bellman Equation
- RL Algorithms
  - Dynamic Programming
  - Policy Iteration
  - Value Iteration
  - Monte Carlo Methods
  - Temporal Difference Learning
  - Multi-step Bootstrapping
  - ……
  - …….
  - Policy Gradient Methods
  - ….
  - RL in Multi Agent Scenarios
    - Game Theory
      - Nash Equilibrium

-Vishal Kumar

dreamerkumar.com

REFERENCES

- Reinforcement Learning: An Introduction
  - By Richard S. Sutton and Andrew G. Barto
- Reinforcement Learning Course by David Silver (YouTube recordings of his lectures at UCL)

-Vishal Kumar
dreamerkumar.com