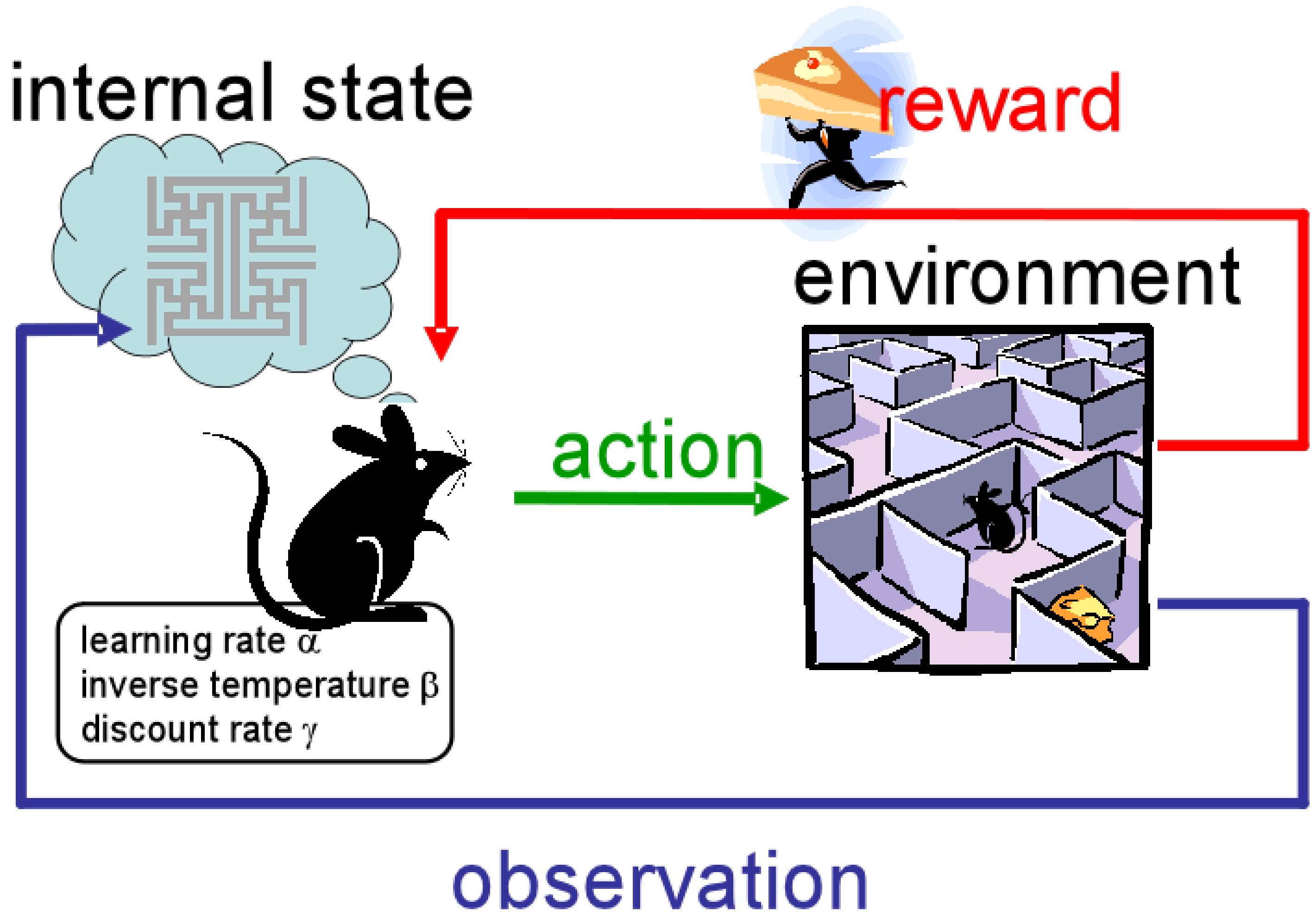



COMP 141: Probabilistic Robotics for Human-Robot Interaction

Instructor: Jivko Sinapov
www.cs.tufts.edu/~jsinapov

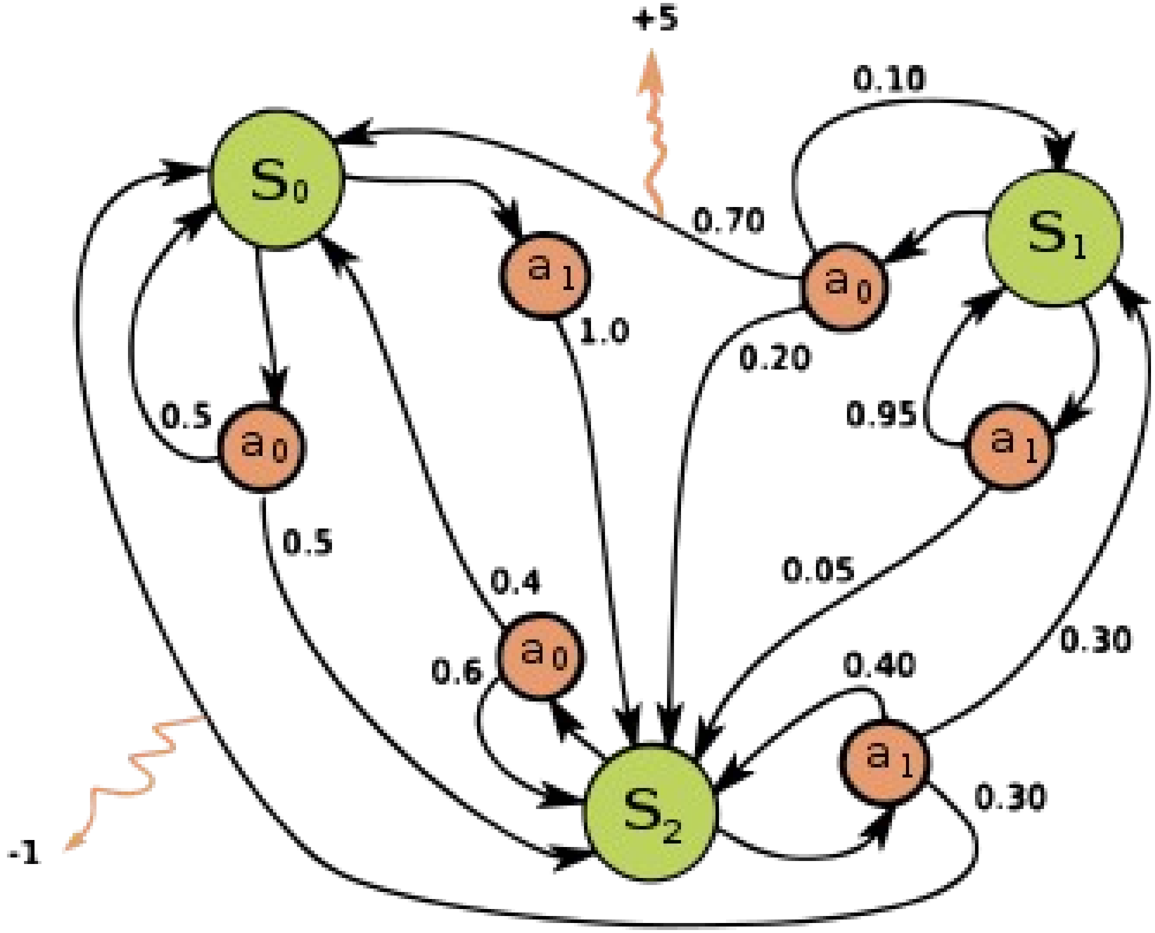
Today: Reinforcement Learning



Announcements

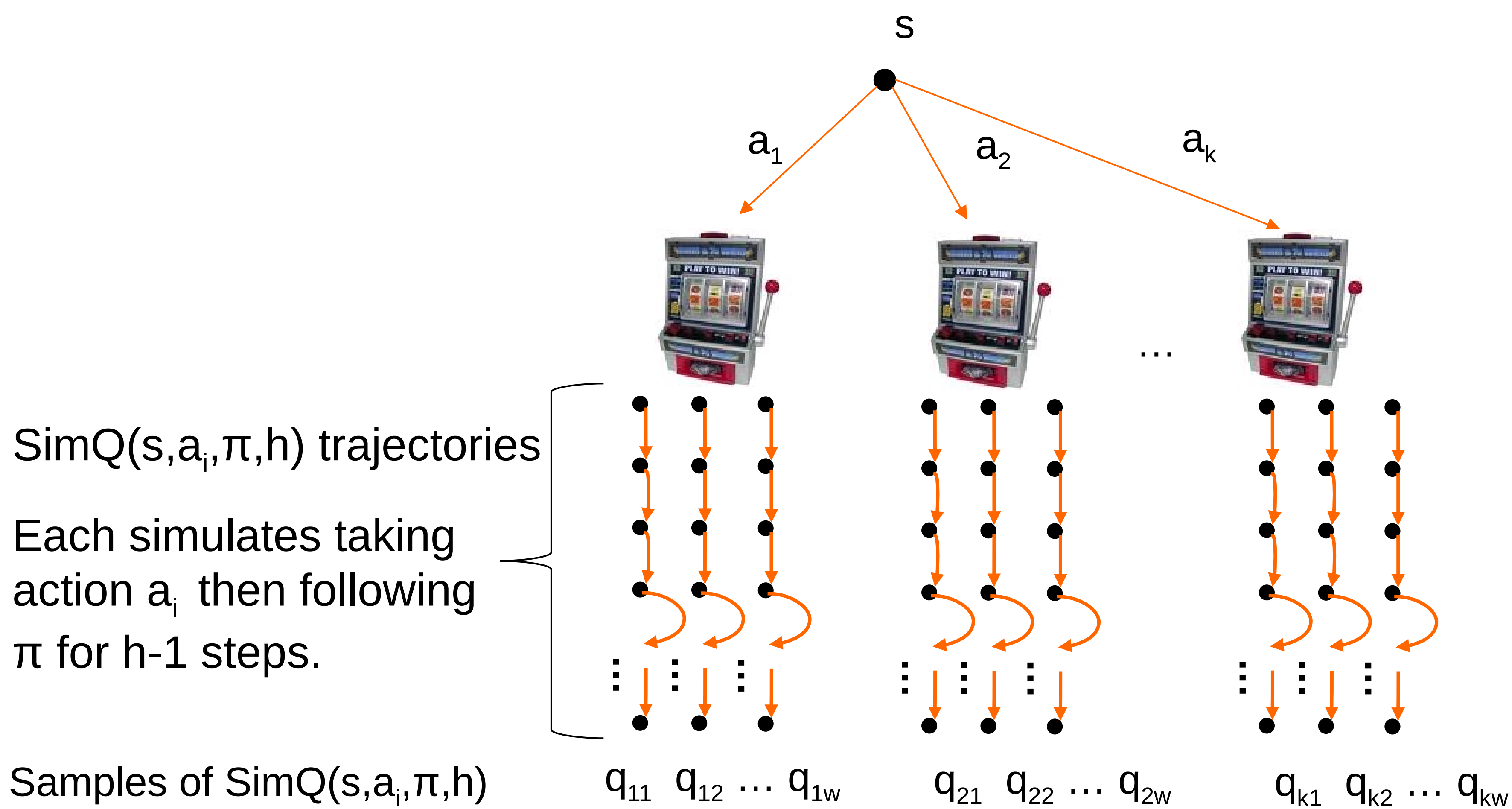
Reading Assignments

Monte Carlo Planning Demo



Policy Rollout Algorithm

1. For each a_i run SimQ(s,a_i,π,h) **w** times
2. Return action with best average of SimQ results



Policy Rollout Algorithm

SimQ(s,a,π,h)

  r = R(s,a)
  s = T(s,a)

simulate a in s

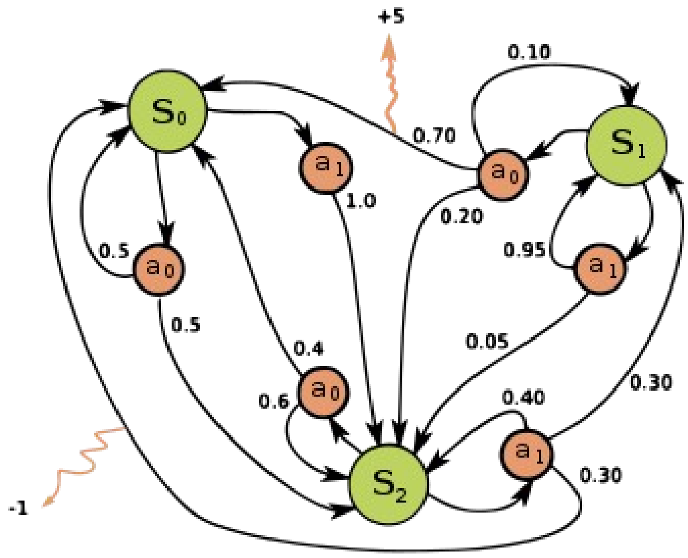
  for i = 1 to h-1
    $r = r + \beta^i R(s, \pi(s))$
    $s = T(s, \pi(s))$

simulate h-1 steps of policy

  Return r

- Simply simulate taking **a** in **s** and following policy for h-1 steps, returning discounted sum of rewards
- Expected value of SimQ(s,a,π,h) is $Q_\pi(s,a,h)$

Python Demo of Policy Improvement



S0	a0	27.16
S0	a1	27.81
S1	a0	29.83
S1	a1	27.66
S2	a0	25.77
S2	a1	25.94

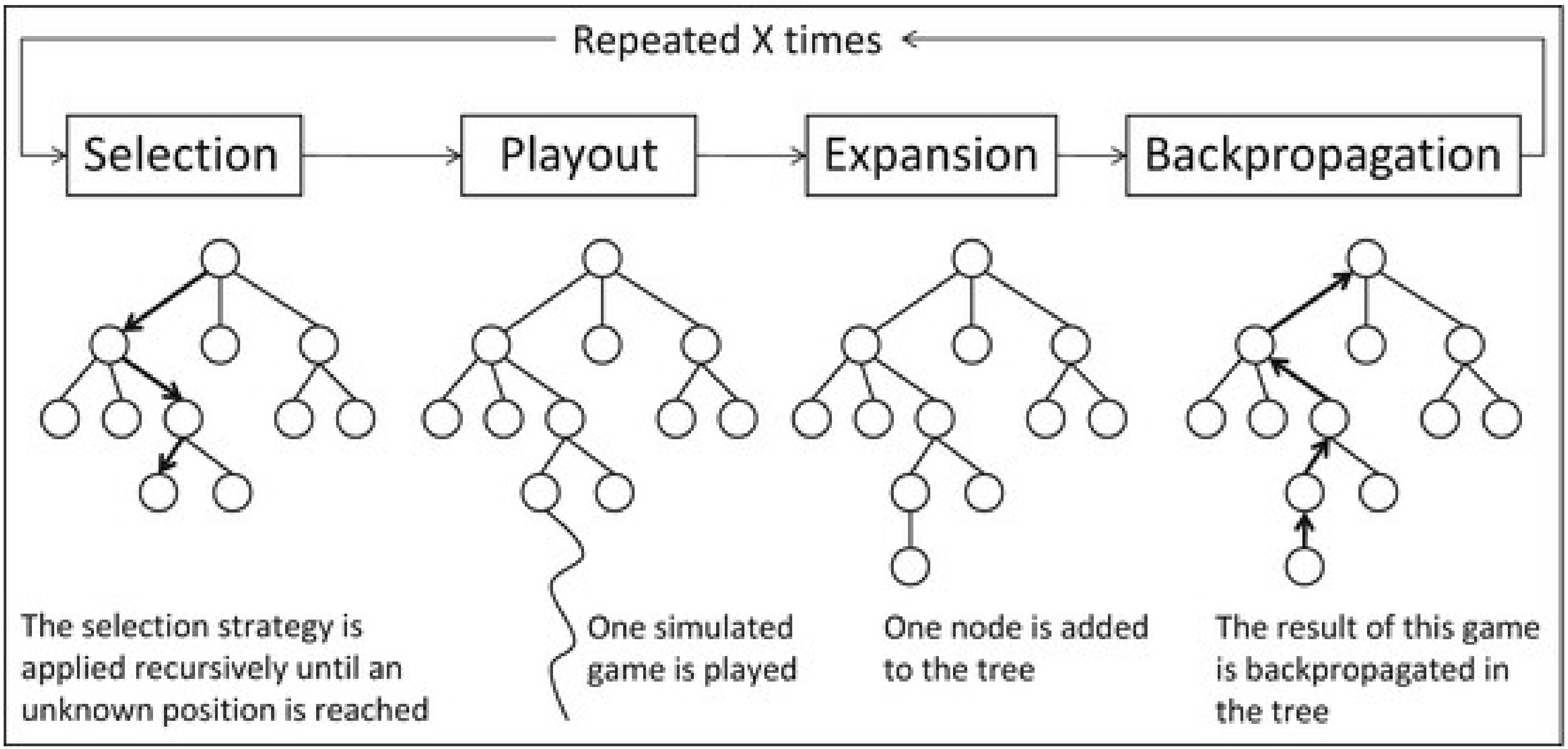
S0	a0	46.67
S0	a1	47.37
S1	a0	48.86
S2	a1	47.14
S3	a0	44.29
S3	a1	44.95

Horizon = 30 time steps
Discount factor = 1.0

State-action value table with random policy

State-action value table with policy derived from table 1

Monte Carlo Tree Search



https://miro.medium.com/max/2982/1*Ntm0xHhJ5jOgsL9AdB2kNw.jpeg

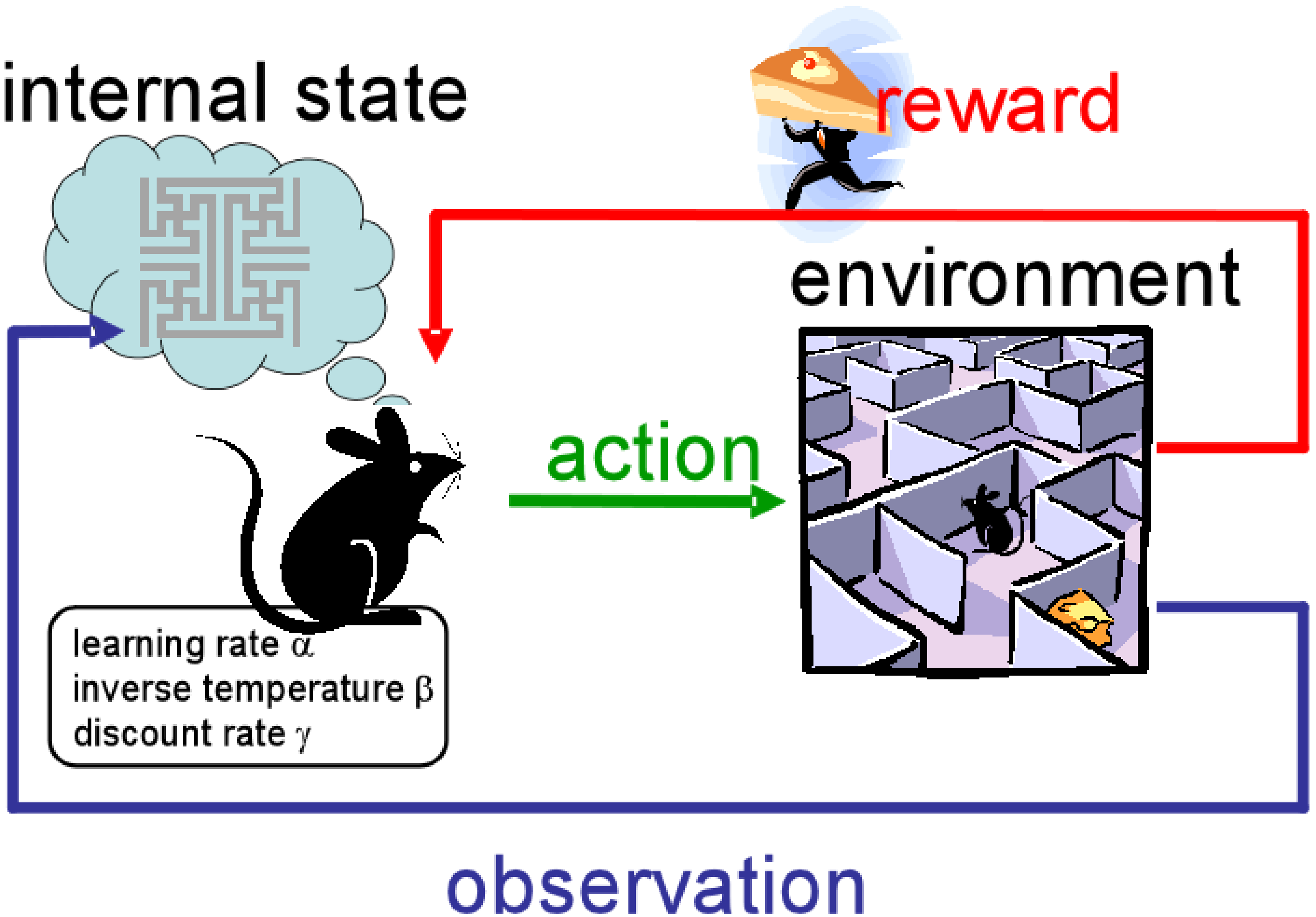
Applications

- https://www.youtube.com/watch?v=NscV-RpRSRw

- https://www.youtube.com/watch?v=gDG1WgZqGKU

- https://www.youtube.com/watch?v=fS5tTa_Tl1Y

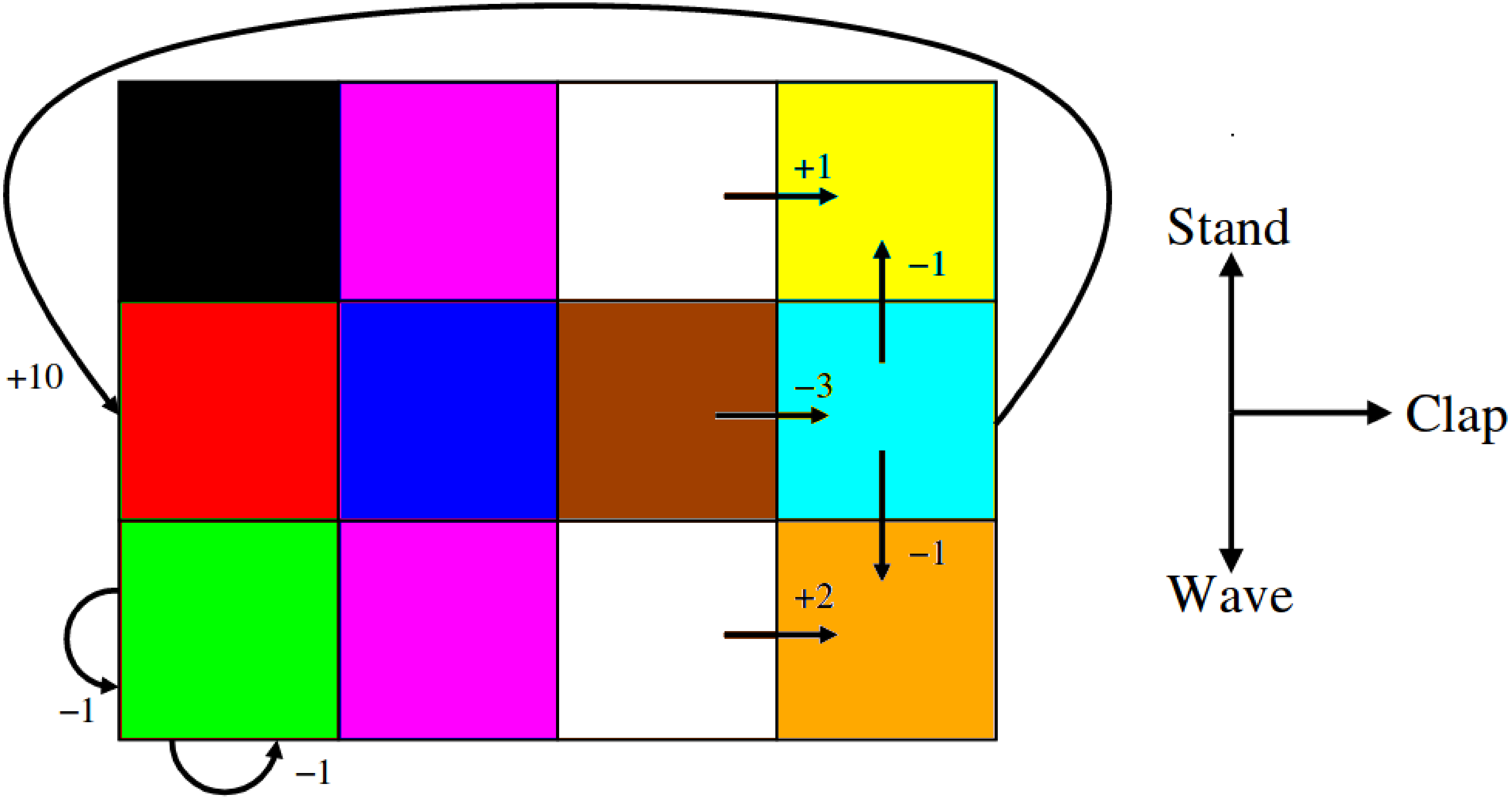
Reinforcement Learning



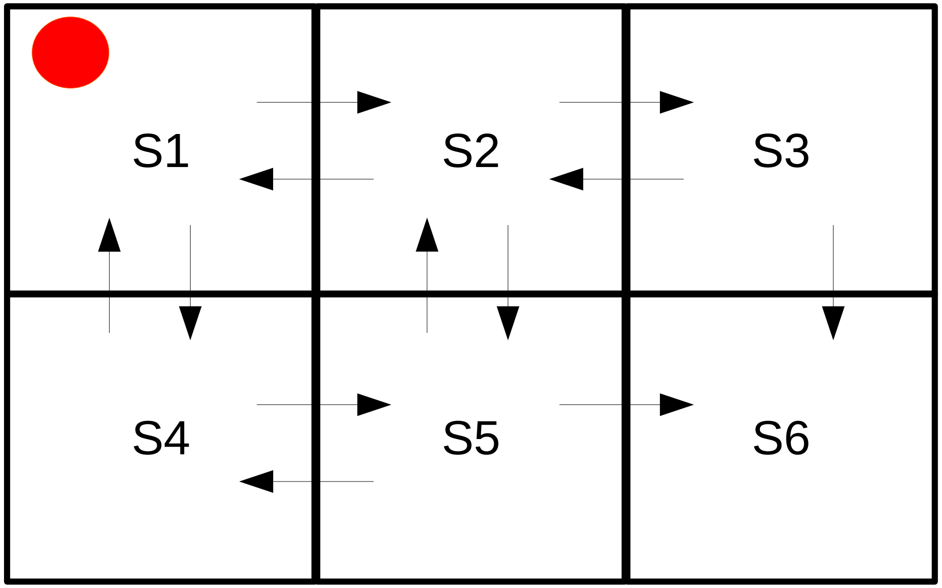
BE a reinforcement learner

- You, as a class, will act as the learning agent
- **Actions:** wave, clap, or nod
- **Observations:** color, reward
- **Goal:** find an optimal *policy*
  - What is a policy? What makes a policy optimal?

What actually happened...



A Simple RL Algorithm: Q-learning



+ 100 reward for getting to S6
0 for all other transitions

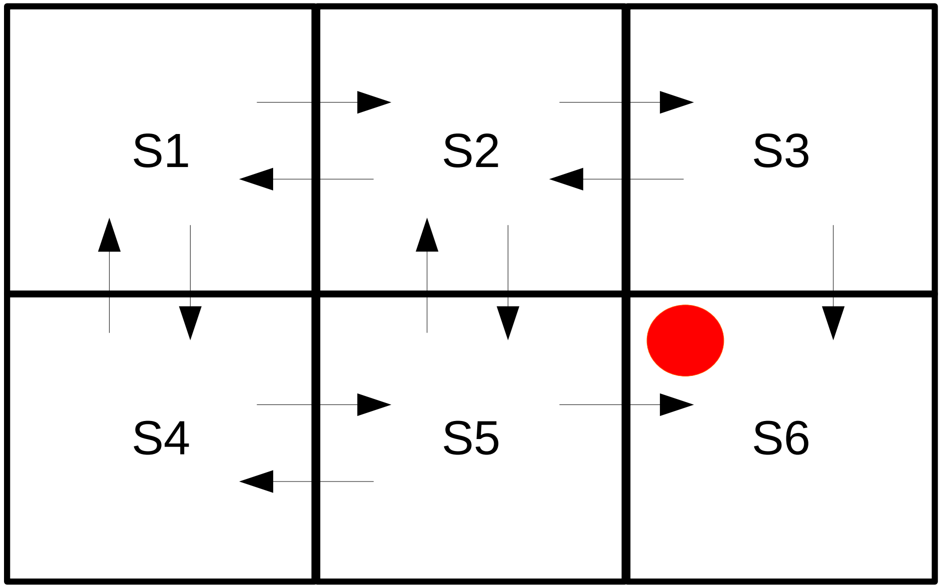
Update rule upon executing action a, ending up in state s' and observing reward r :

$$Q(s\ ,\ a)=r\ +\gamma\ \max a'\ Q(s'\ ,a'\ )$$

$$\gamma = 0.5 \text{ (discount factor)}$$

Q-Table

S1	right	
S1	down	
S2	right	
S2	left	
S2	down	
S3	left	
S3	down	
S4	up	
S4	right	
S5	left	
S5	up	
S5	right	



+ 100 reward for getting to S6
0 for all other transitions

Update rule upon executing action a, ending up in state s'
and observing reward r :

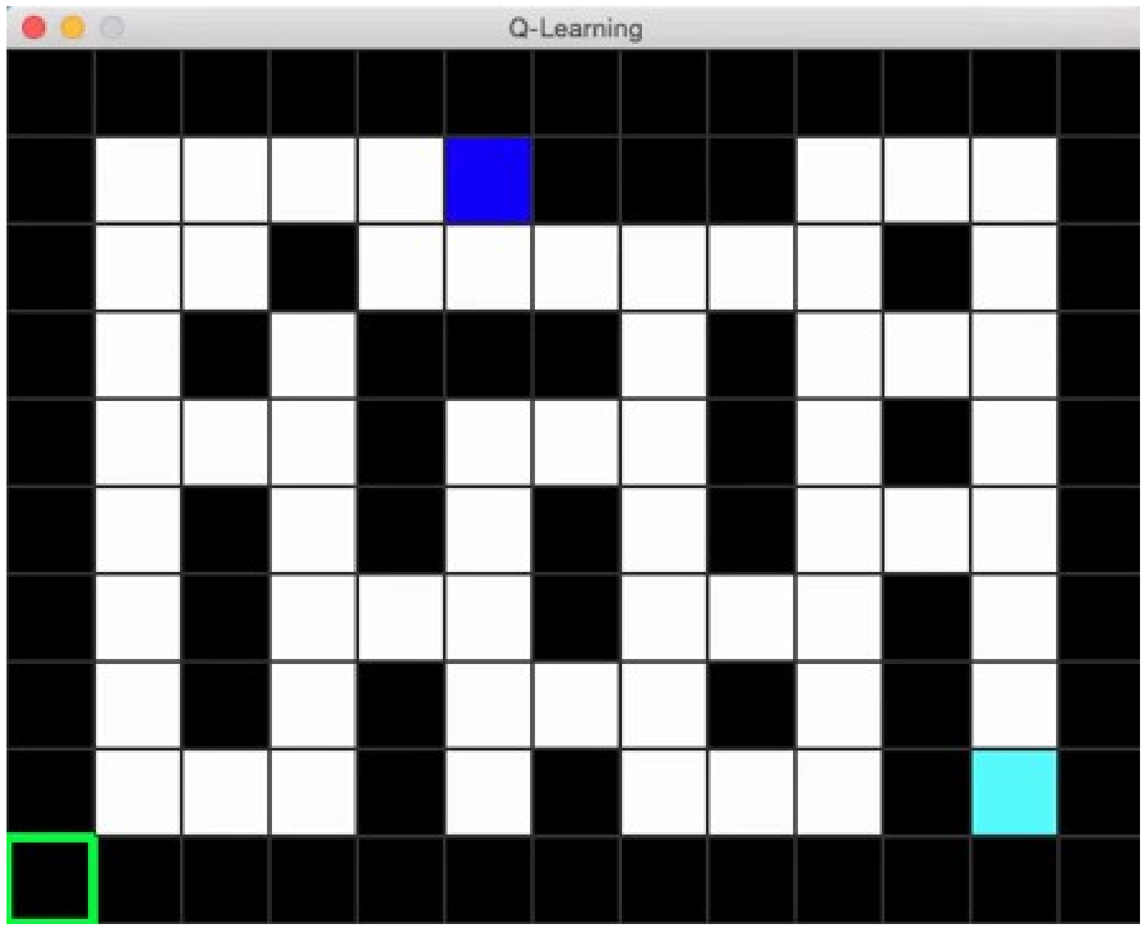
$$Q(s\ ,\ a) = r + \gamma \max_{a'} Q(s'\ , a')$$

$$\gamma = 0.5 \text{ (discount factor)}$$

Q-Table

S1	right	25
S1	down	25
S2	right	50
S2	left	12.5
S2	down	50
S3	left	25
S3	down	100
S4	up	12.5
S4	right	50
S5	left	25
S5	up	25
S5	right	100

Example with larger board



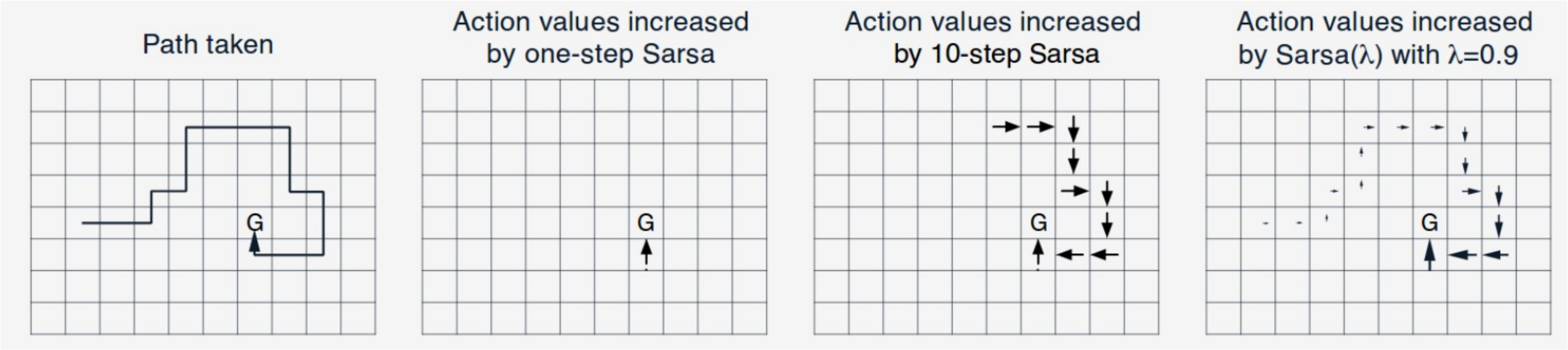
Q-Learning Algorithm

Q-learning (off-policy TD control) for estimating $\pi \approx \pi_*$

Initialize $Q(s, a)$, for all $s \in \mathcal{S}, a \in \mathcal{A}(s)$, arbitrarily, and $Q(\textit{terminal-state}, \cdot) = 0$
Repeat (for each episode):
    Initialize S
    Repeat (for each step of episode):
        Choose A from S using policy derived from Q (e.g., ϵ-greedy)
        Take action A, observe R, S'
        $Q(S, A) \leftarrow Q(S, A) + \alpha\big[R + \gamma \max_a Q(S', a) - Q(S, A)\big]$
        $S \leftarrow S'$
    until S is terminal

[Sutton and Barto]

Other Approaches



Some Applications of RL in Robotics

https://www.youtube.com/watch?v=W_gxLKSsSIE&t=12s

https://www.youtube.com/watch?v=ZVIxt2rt1_4

https://www.youtube.com/watch?v=QZvu8M02BeE

https://www.youtube.com/watch?v=mRpX9DFCdwI

If you want to know more about RL...

- take CS 138: Reinforcement Learning in the Fall

Student Paper Presentation