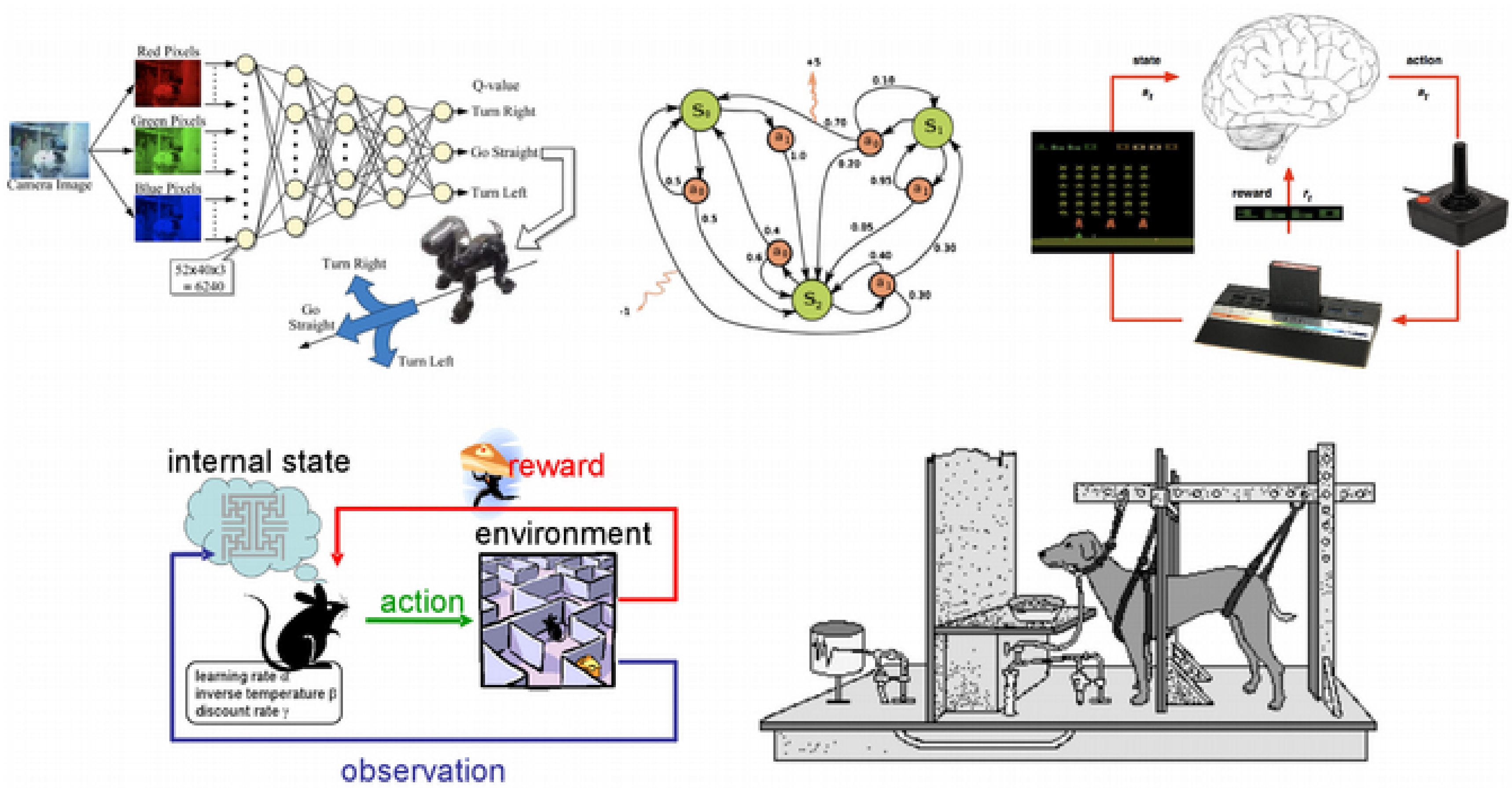
COMP 138: Reinforcement Learning



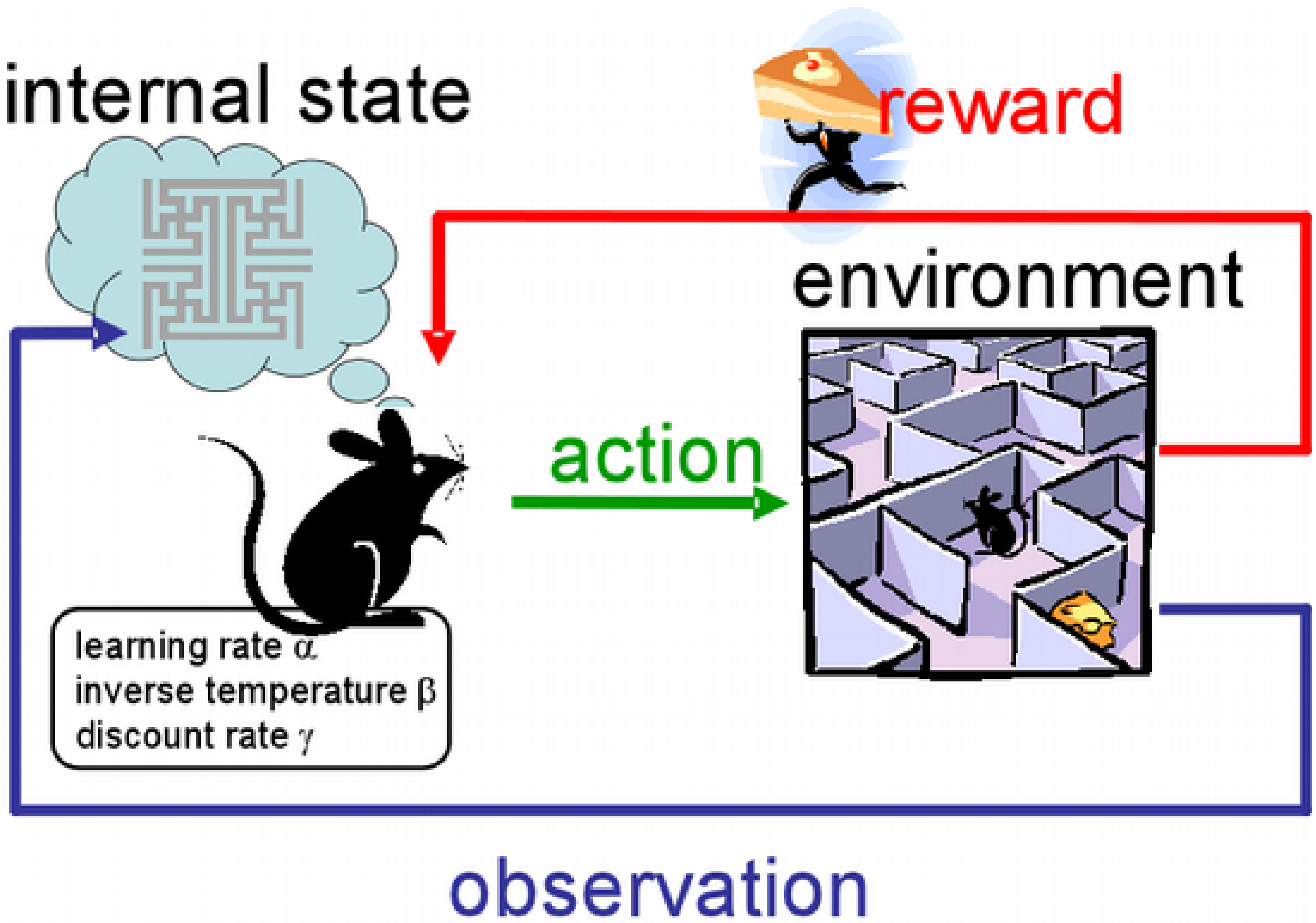
Instructor: Jivko Sinapov
Webpage: https://www.eecs.tufts.edu/~jsinapov/teaching/comp150_RL_Fall2020/

Announcements

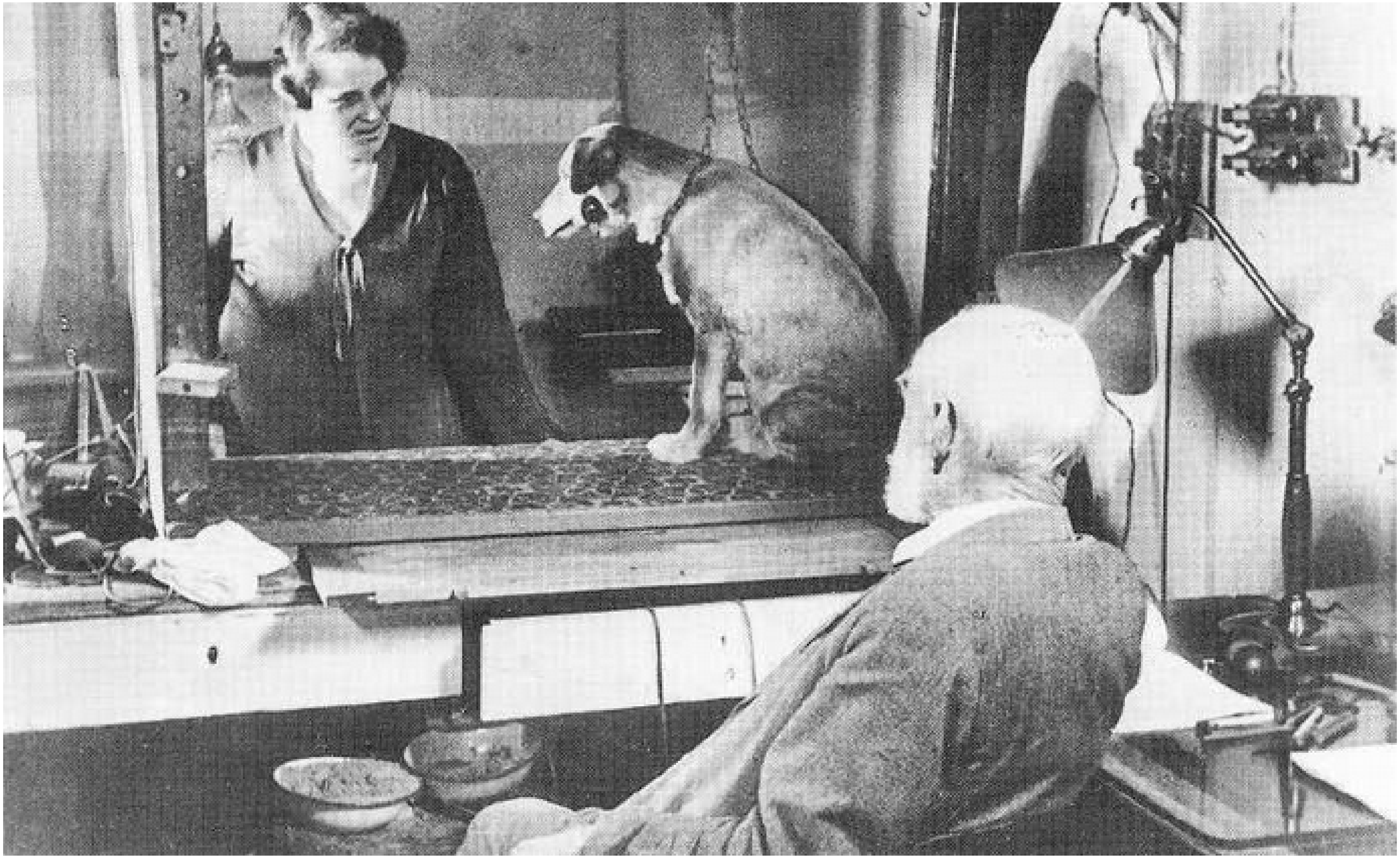
CS Colloquium Today

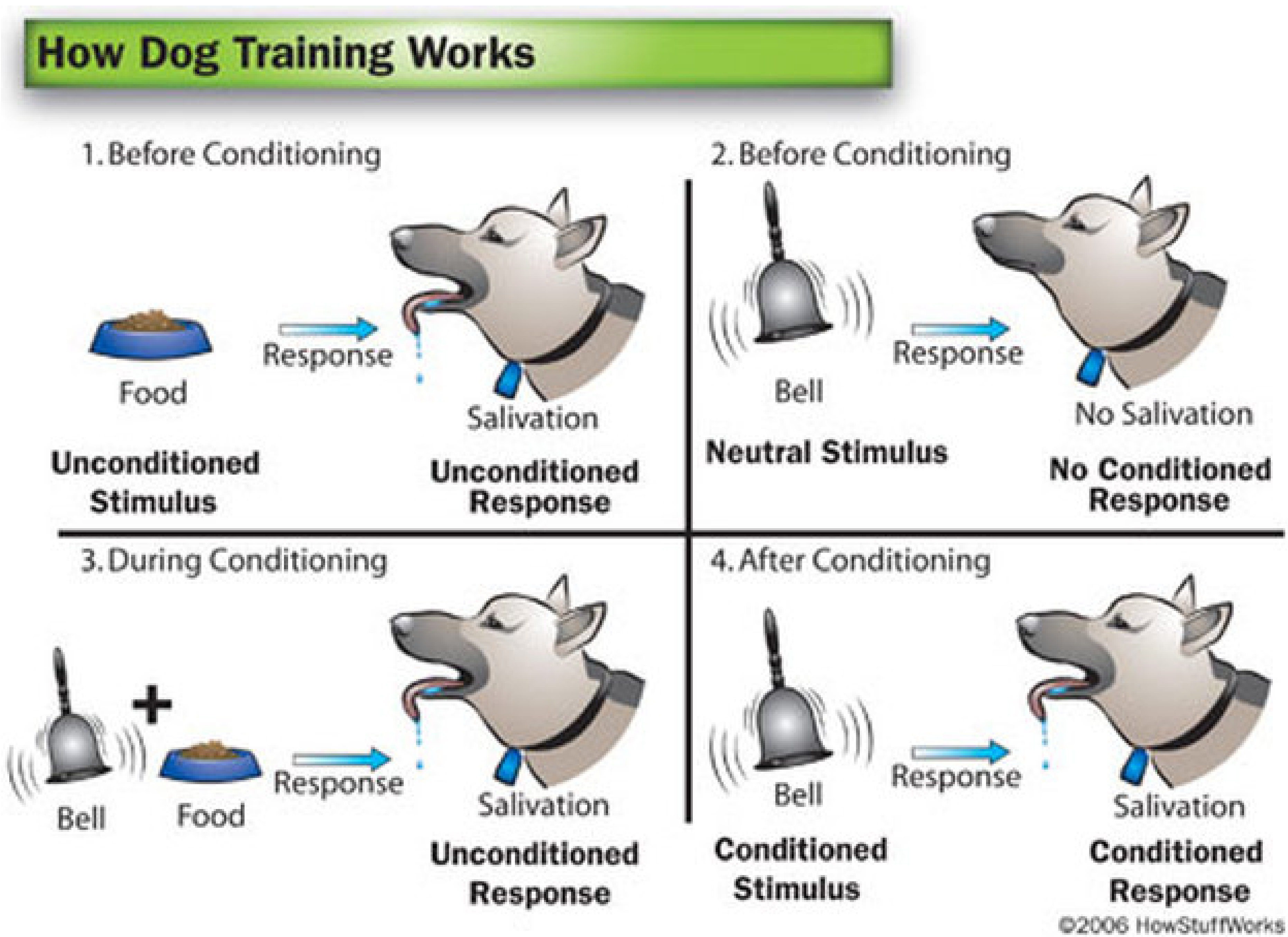
- Title: Autonomous Vehicles and Persons with Disabilities: Current Work and Future Directions
- Speaker: Julian Brinkley, Clemson University
- 3:00 pm @ Virtual Halligan 102

Reinforcement Learning

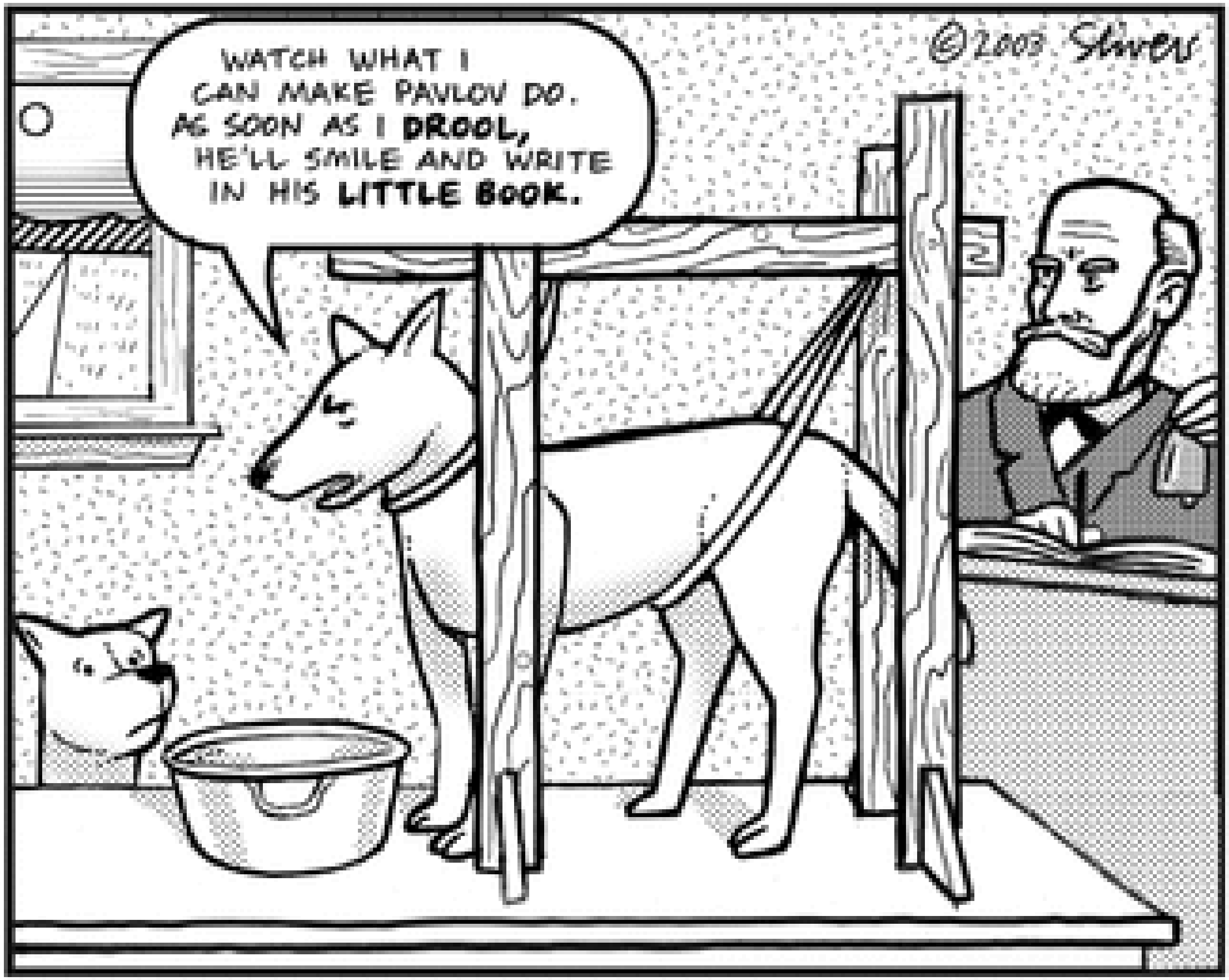


Ivan Pavlov (1849-1936)





How Dog Training Works
1. Before Conditioning
Food
Response
Salivation
Unconditioned Stimulus
Unconditioned Response
2. Before Conditioning
Bell
Response
No Salivation
Neutral Stimulus
No Conditioned Response
3. During Conditioning
Bell
Food
Response
Salivation
Unconditioned Response
4. After Conditioning
Bell
Response
Salivation
Conditioned Stimulus
Conditioned Response
©2006 HowStuffWorks

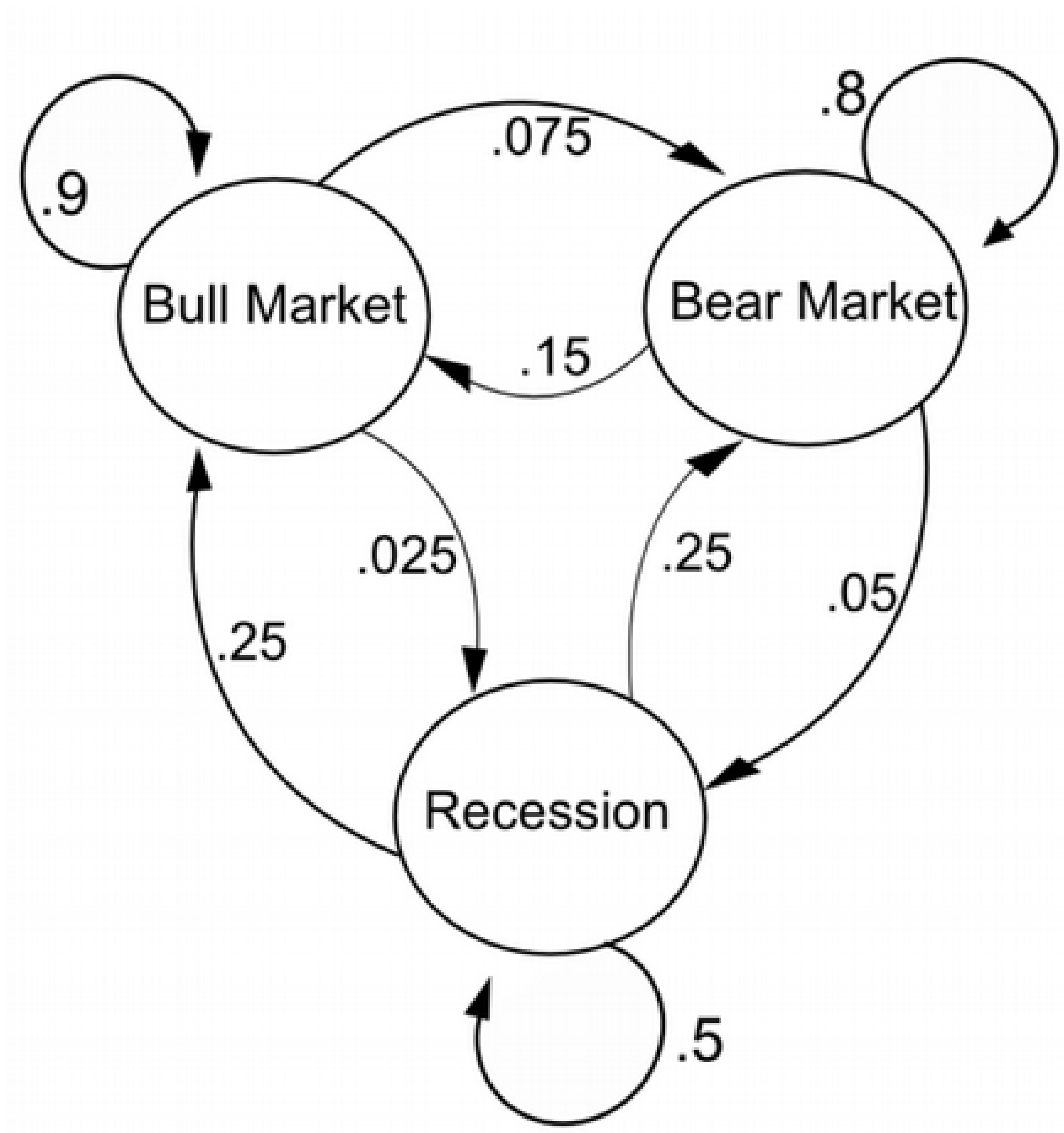


WATCH WHAT I CAN MAKE PAVLOV DO. AS SOON AS I DROOL, HE'LL SMILE AND WRITE IN HIS LITTLE BOOK.
©2003 Stivers

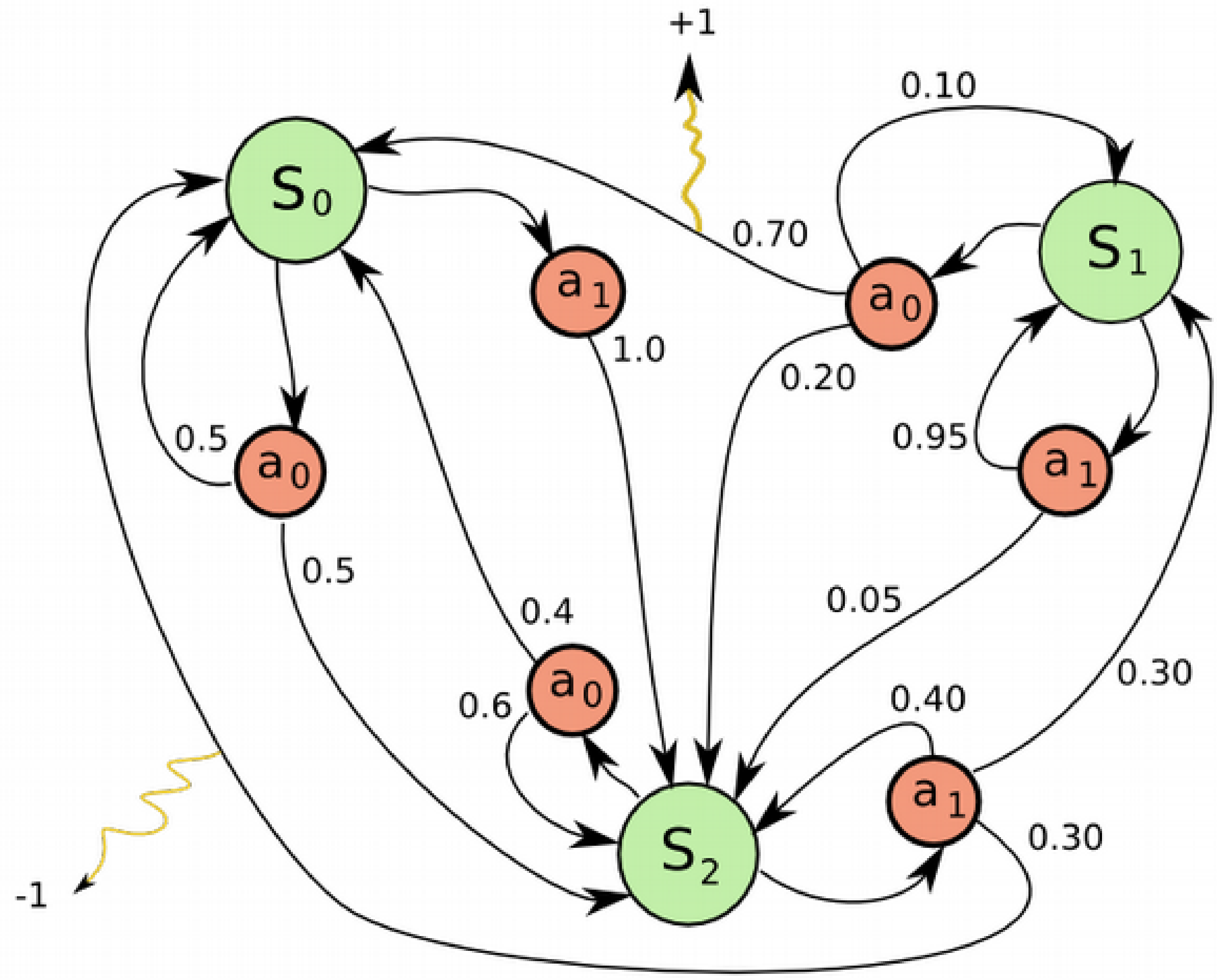
Andrey Andreyevich Markov (1856 – 1922)

[http://en.wikipedia.org/wiki/Andrey_Markov]

Markov Chain



Markov Decision Process



The Multi-Armed Bandit Problem

a.k.a. how to pick between Slot Machines (one-armed bandits) so that you walk out with the most $$$ from the Casino

Arm 1

Arm 2

. . . .

Arm k

How should we decide which slot machine to pull next?

How should we decide which slot machine to pull next?

0 1 3 0 1

0 0 0 50 0

How should we decide which slot machine to pull next?

1 with prob = 0.6 and 0 otherwise

50 with prob = 0.01 and 0 otherwise

Value Function

A value function encodes the "value" of performing a particular action (i.e., bandit)

Rewards observed when performing action *a*

$$Q_t(a) = \frac{R_1 + R_2 + \cdots + R_{K_a}}{K_a}.$$

Value function Q

# of times the agent has picked action *a*

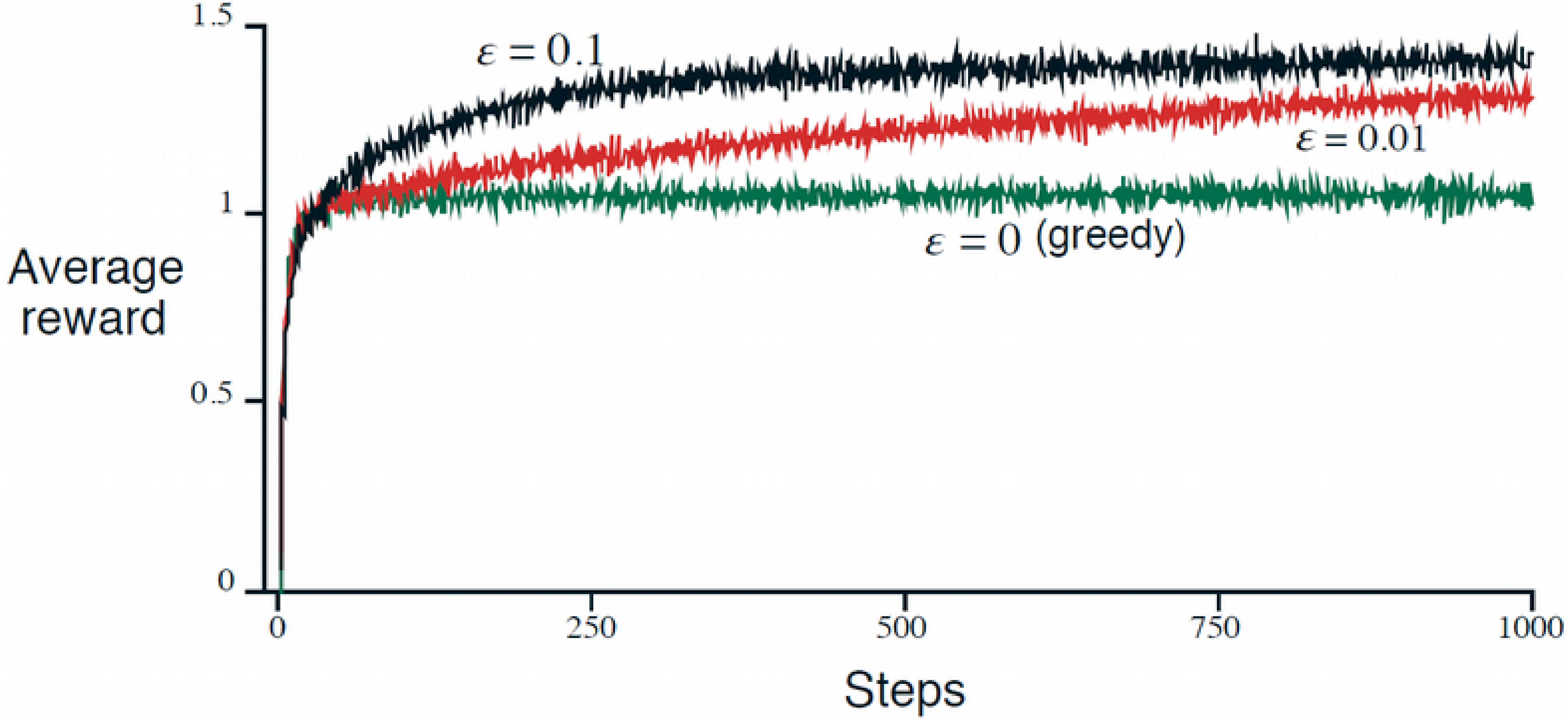
How do we choose next action?

- Greedy: pick the action that maximizes the value function, i.e.,

$$Q_t(A_t^*) = \max_a Q_t(a)$$

- ε-Greedy: with probability ε pick a random action, otherwise, be greedy

10-armed Bandit Example



Soft-Max Action Selection

Exponent of natural logarithm (~ 2.718)

$$\frac{e^{Q_t(a)/\tau}}{\sum_{i=1}^{n} e^{Q_t(i)/\tau}}$$

"temperature"

As temperature goes up, all actions become nearly equally likely to be selected; as it goes down, those with higher value function outputs become more likely

What happens after choosing an action?

Batch:

$$Q_t(a) = \frac{R_1 + R_2 + \cdots + R_{K_a}}{K_a}$$

Incremental:

$$\begin{aligned} Q_{k+1} &= \frac{1}{k}\sum_{i=1}^{k} R_i \\ &= \frac{1}{k}\left(R_k + \sum_{i=1}^{k-1} R_i\right) \\ &= \frac{1}{k}\Big(R_k + (k-1)Q_k + Q_k - Q_k\Big) \\ &= \frac{1}{k}\Big(R_k + kQ_k - Q_k\Big) \\ &= Q_k + \frac{1}{k}\Big[R_k - Q_k\Big], \end{aligned}$$

Updating the Value Function

$$NewEstimate \leftarrow OldEstimate + StepSize\left[Target - OldEstimate\right]$$

What happens when the payout of a bandit is changing over time?

$$Q_t(a) = \frac{R_1 + R_2 + \cdots + R_{K_a}}{K_a}$$

What happens when the payout of a bandit is changing over time?

$$Q_t(a) = \frac{R_1 + R_2 + \cdots + R_{K_a}}{K_a}$$

Earlier rewards may not be indicative of
how the bandit performs now

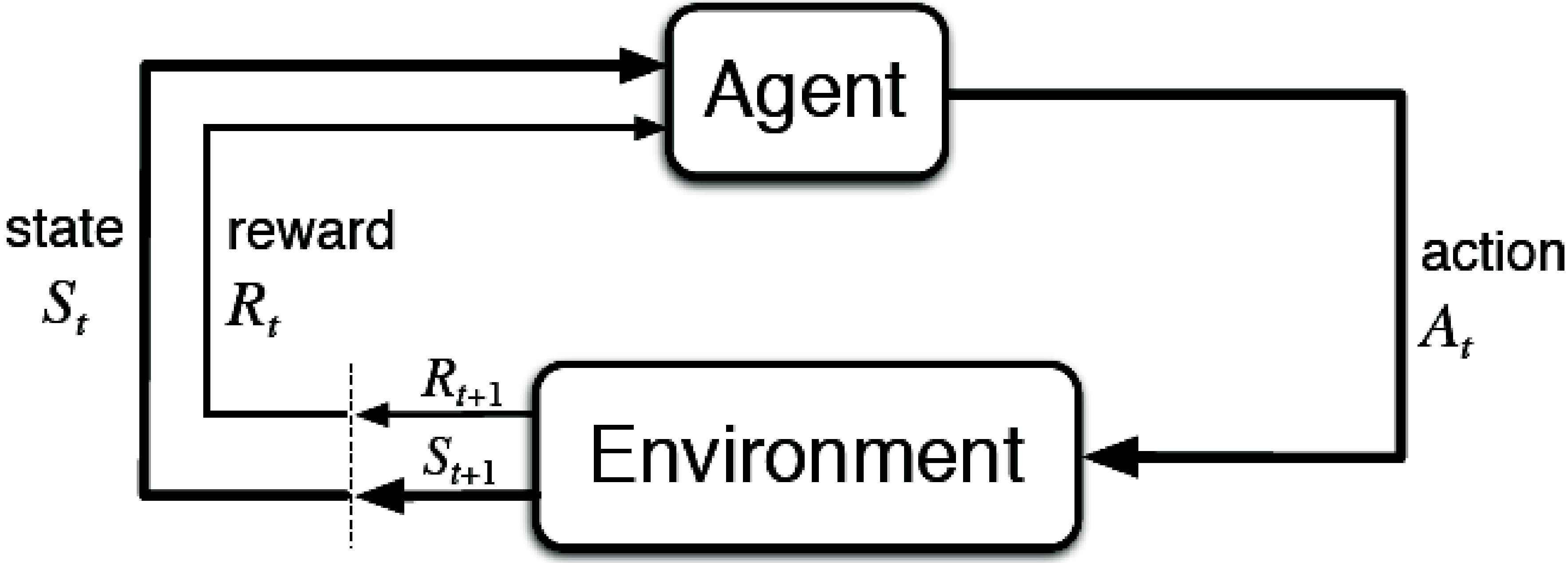
What happens when the payout of a bandit is changing over time?

$$Q_{k+1} = Q_k + \alpha \Big[R_k - Q_k \Big]$$

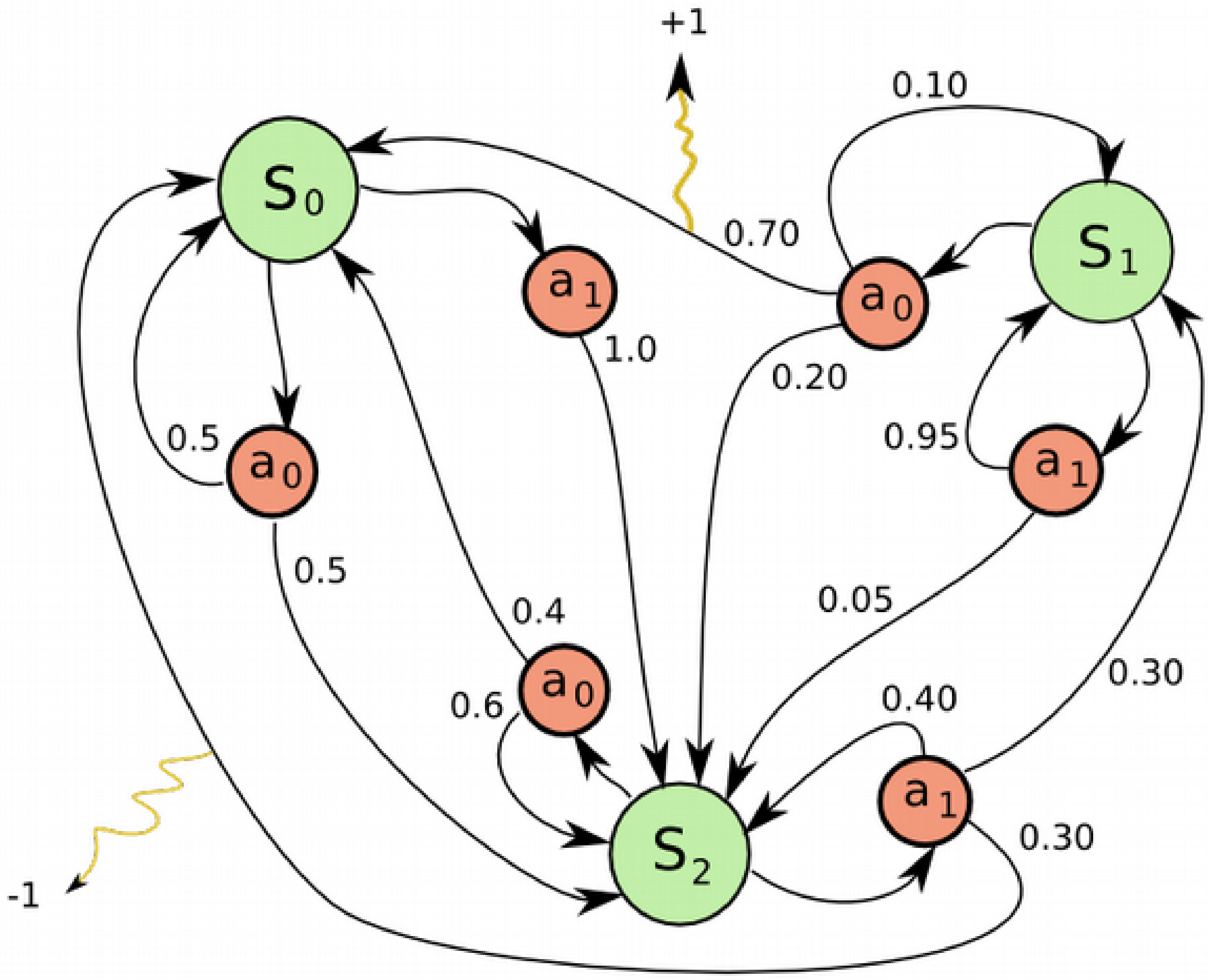
instead of

$$Q_k + \frac{1}{k} \Big[R_k - Q_k \Big]$$

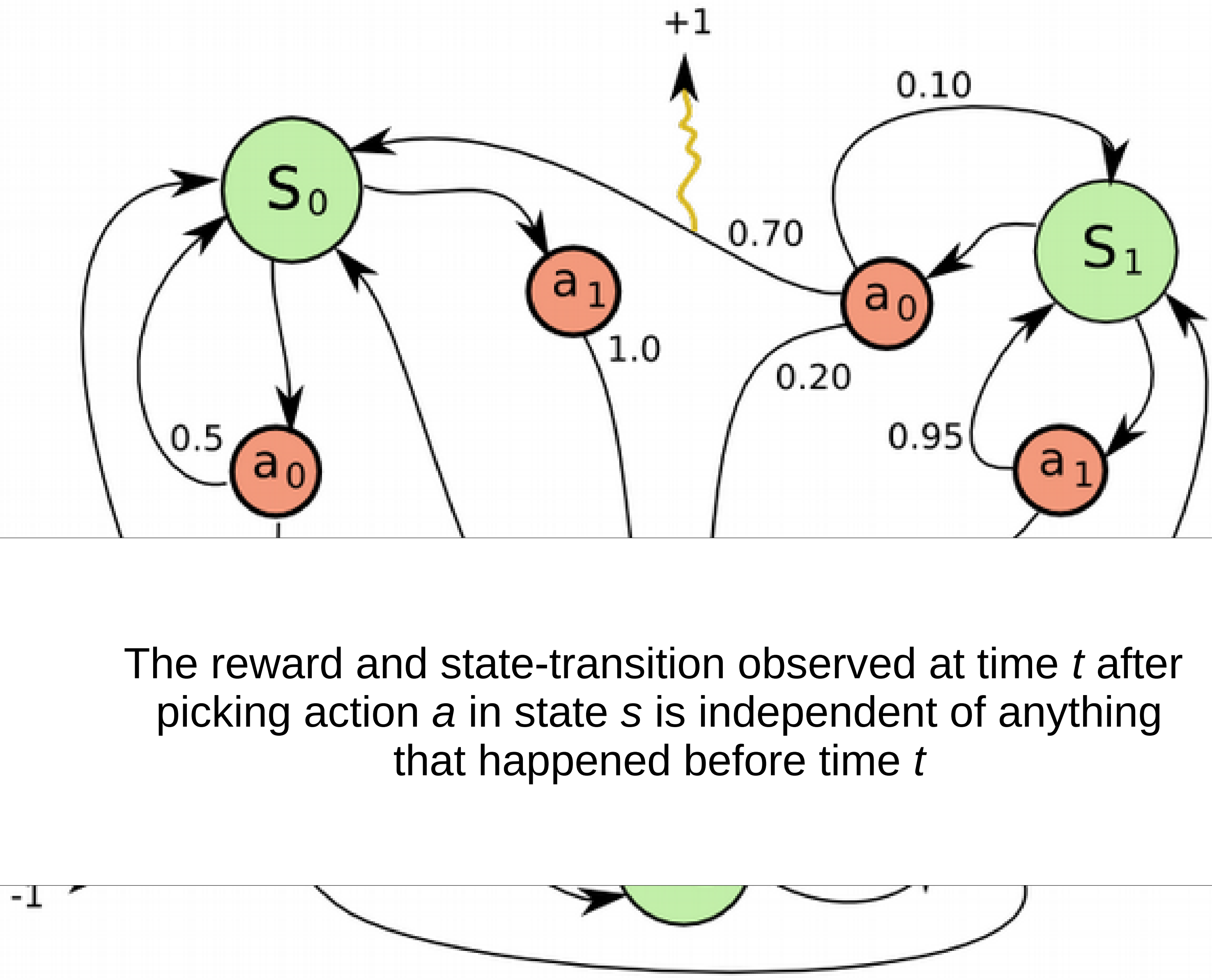
The Reinforcement Learning Problem



RL in the context of MDPs



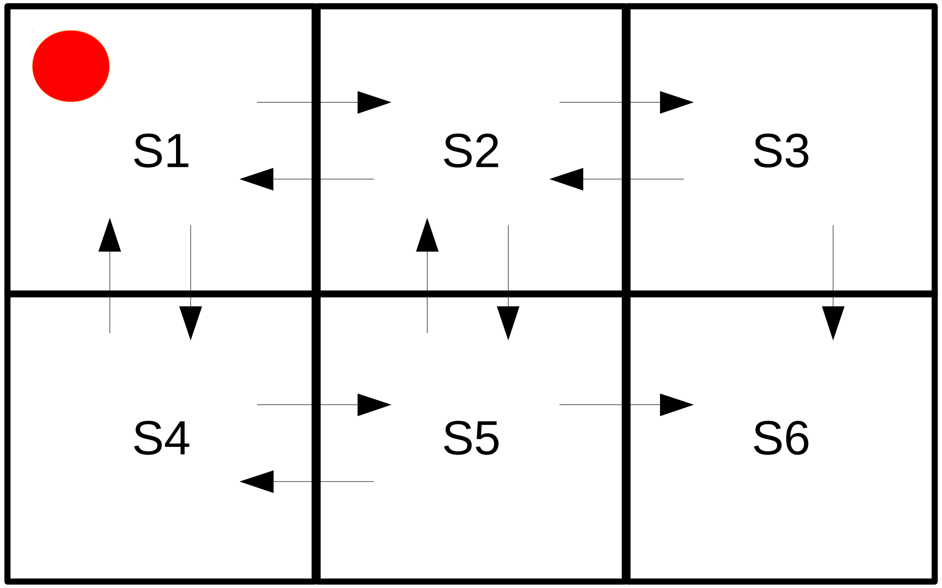
The Markov Assumption



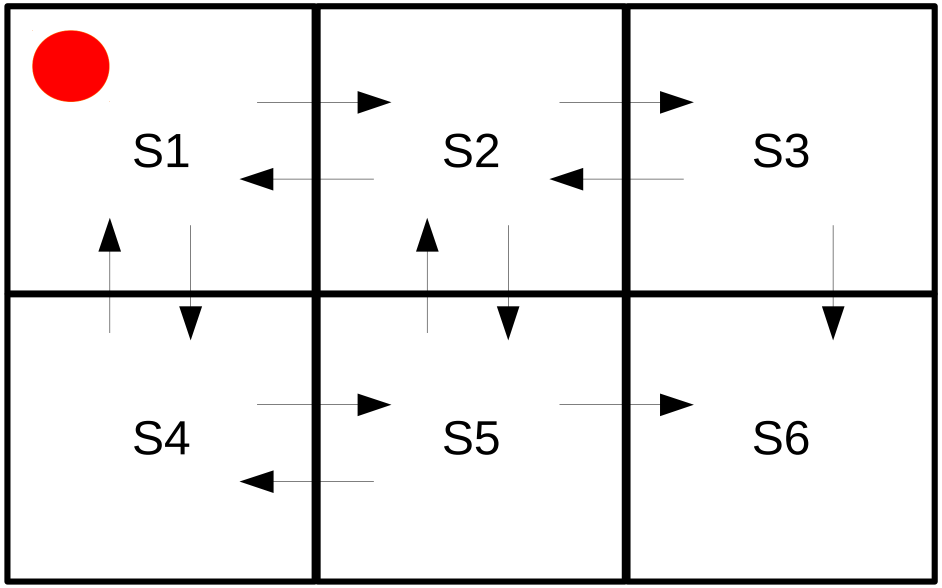
The reward and state-transition observed at time *t* after picking action *a* in state *s* is independent of anything that happened before time *t*

Q-Learning

(board exercise)



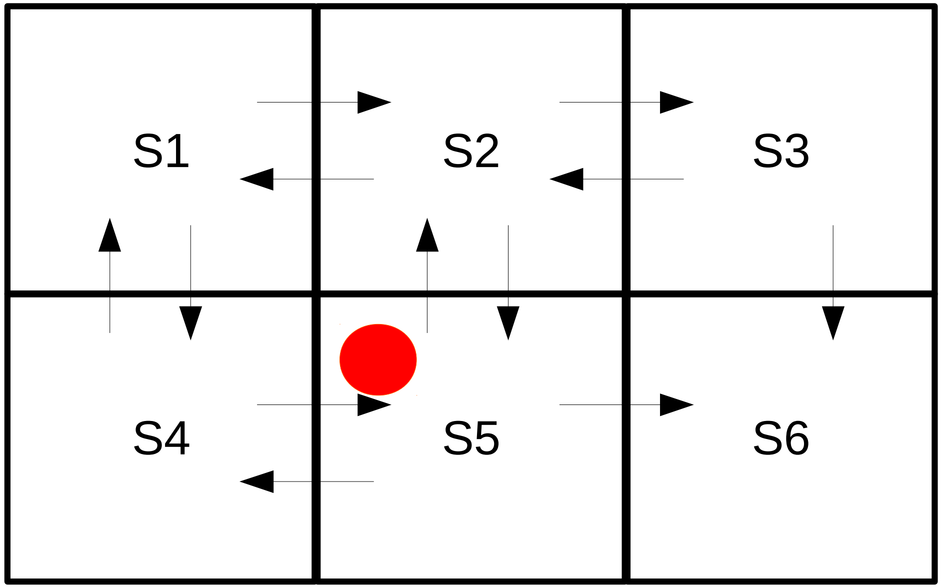
+ 100 reward for getting to S6
0 for all other transitions



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

Q-Table

S1	right	0
S1	down	0
S2	right	0
S2	left	0
S2	down	0
S3	left	0
S3	down	0
S4	up	0
S4	right	0
S5	left	0
S5	up	0
S5	right	0



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

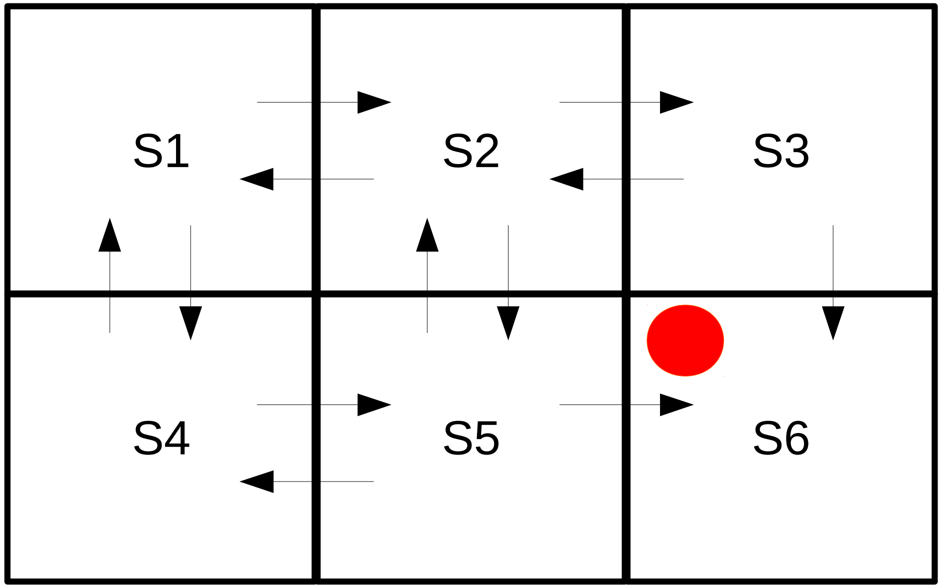
Update rule upon executing action a in state s, ending up in state s' and observing reward r :

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

γ = 0.5 (discount factor)

Q-Table

S1	right	25
S1	down	0
S2	right	50
S2	left	0
S2	down	50
S3	left	0
S3	down	100
S4	up	0
S4	right	0
S5	left	0
S5	up	0
S5	right	100



+ 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 :

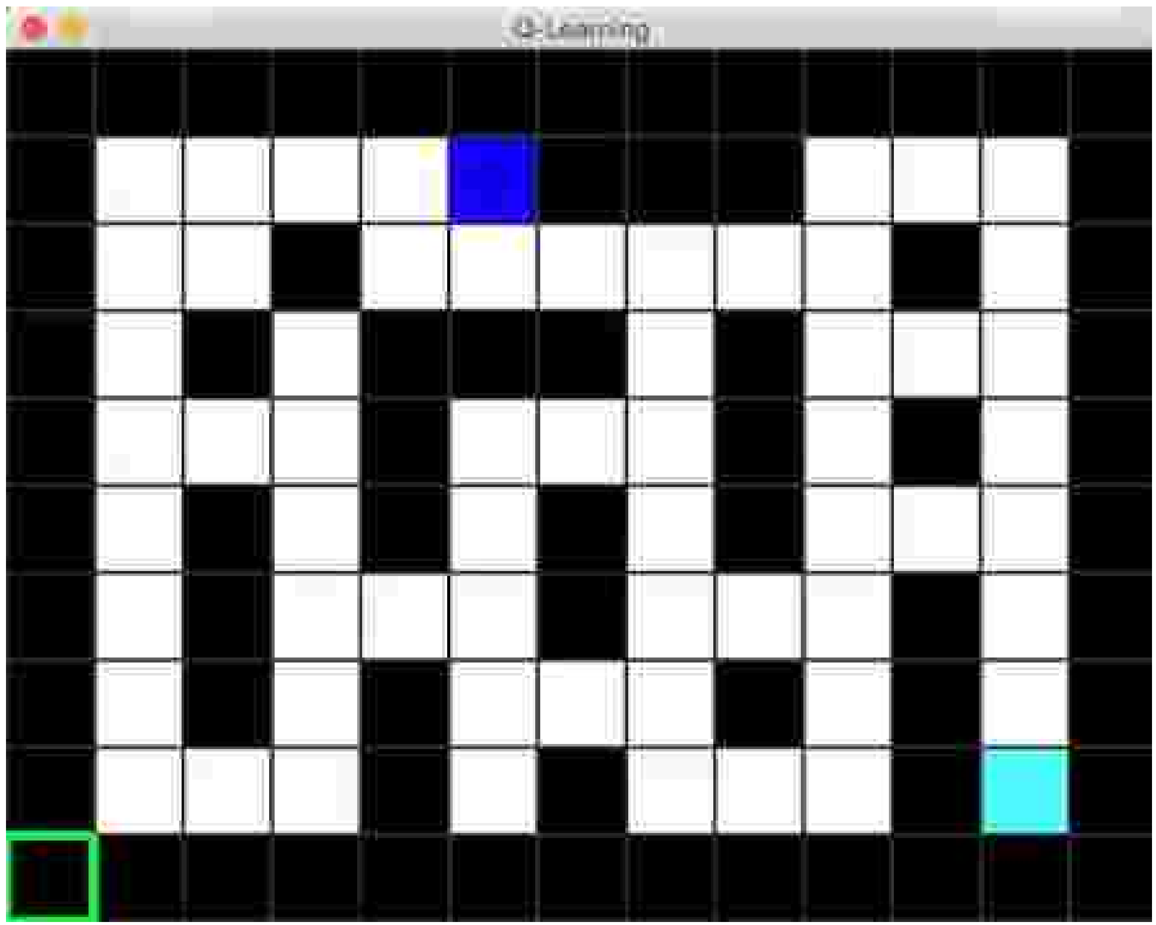
$$Q(s, a) = r + \gamma \max_{a'} Q(s', a')$$

γ = 0.5 (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 a Larger Board



Q-Learning Algorithm

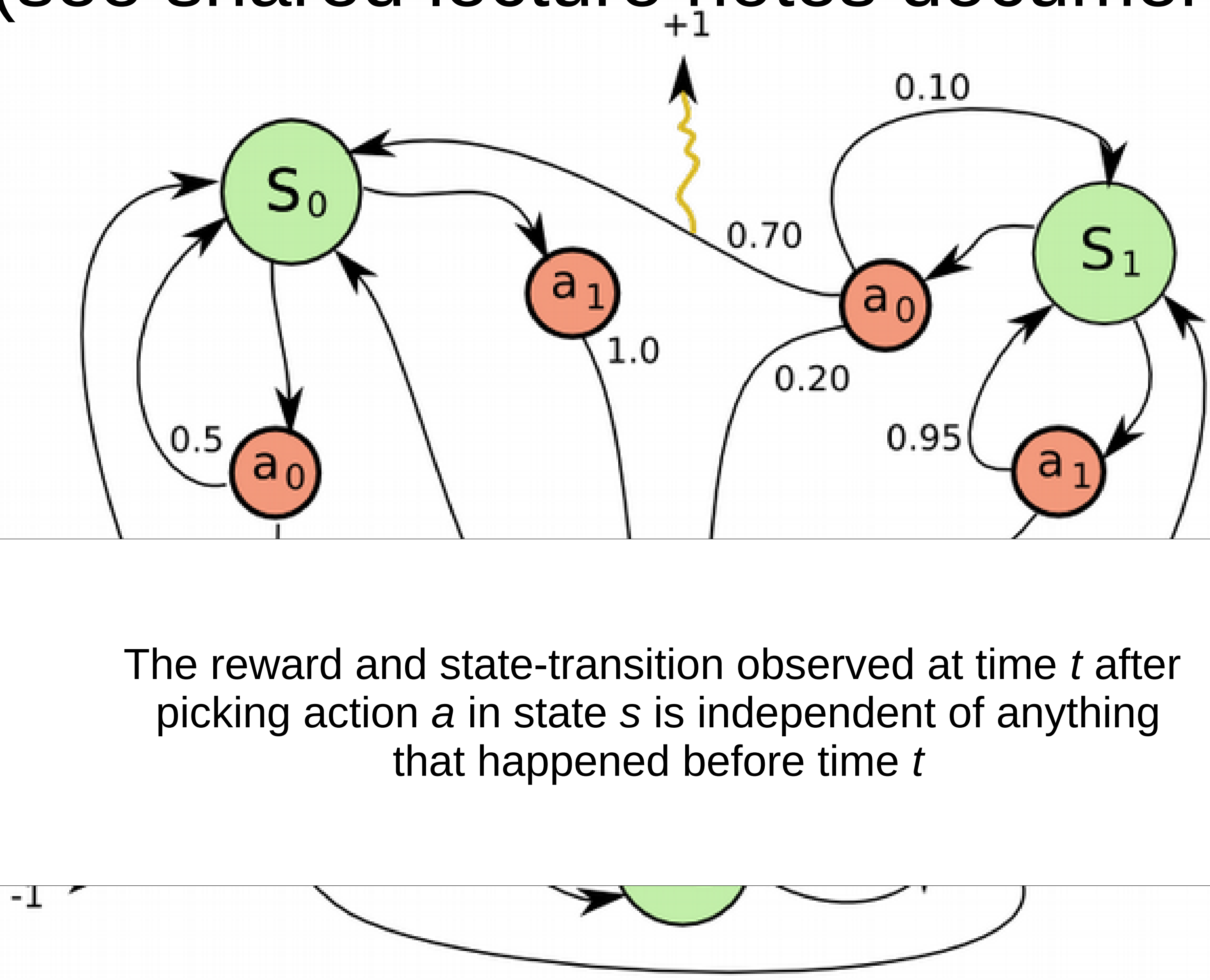
Q-learning (off-policy TD control) for estimating $\pi \approx \pi_*$

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

Q-Learning Properties

- Convergence to the true Q-function is guaranteed...as long as we visit every state-action pair infinitely many times!
- Table size can be very large for complex problems
- We cannot estimate unseen values
- How do we fix these problems?

The Markov Assumption – Exercise
(see shared lecture notes document)



The reward and state-transition observed at time *t* after picking action *a* in state *s* is independent of anything that happened before time *t*

Learning to Shoot Penalty Kicks

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