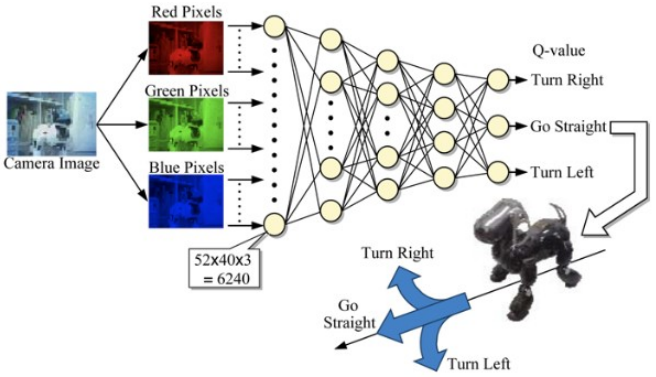
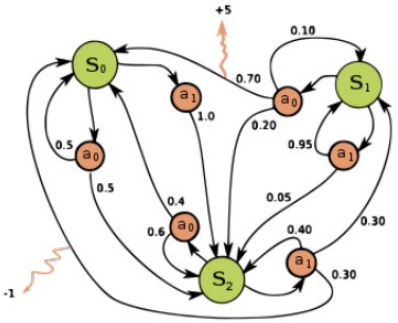
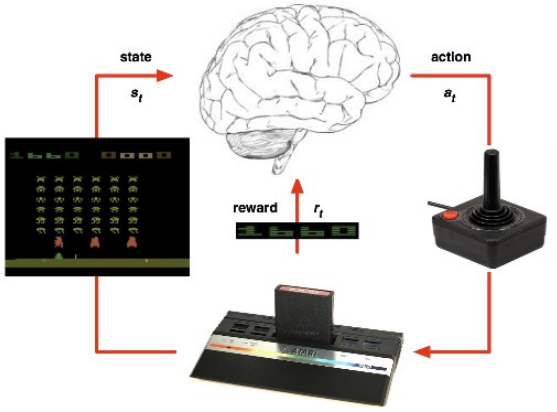
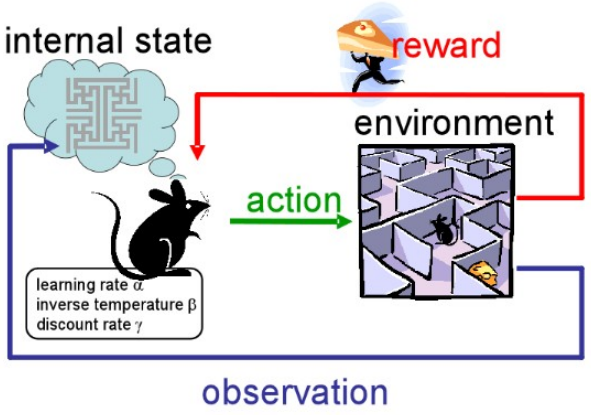
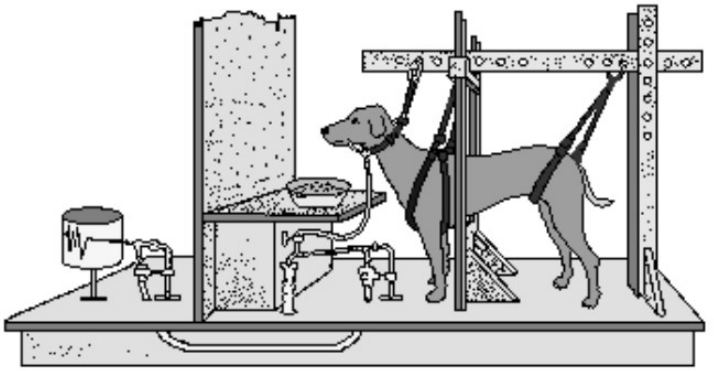
COMP 138: Reinforcement Learning











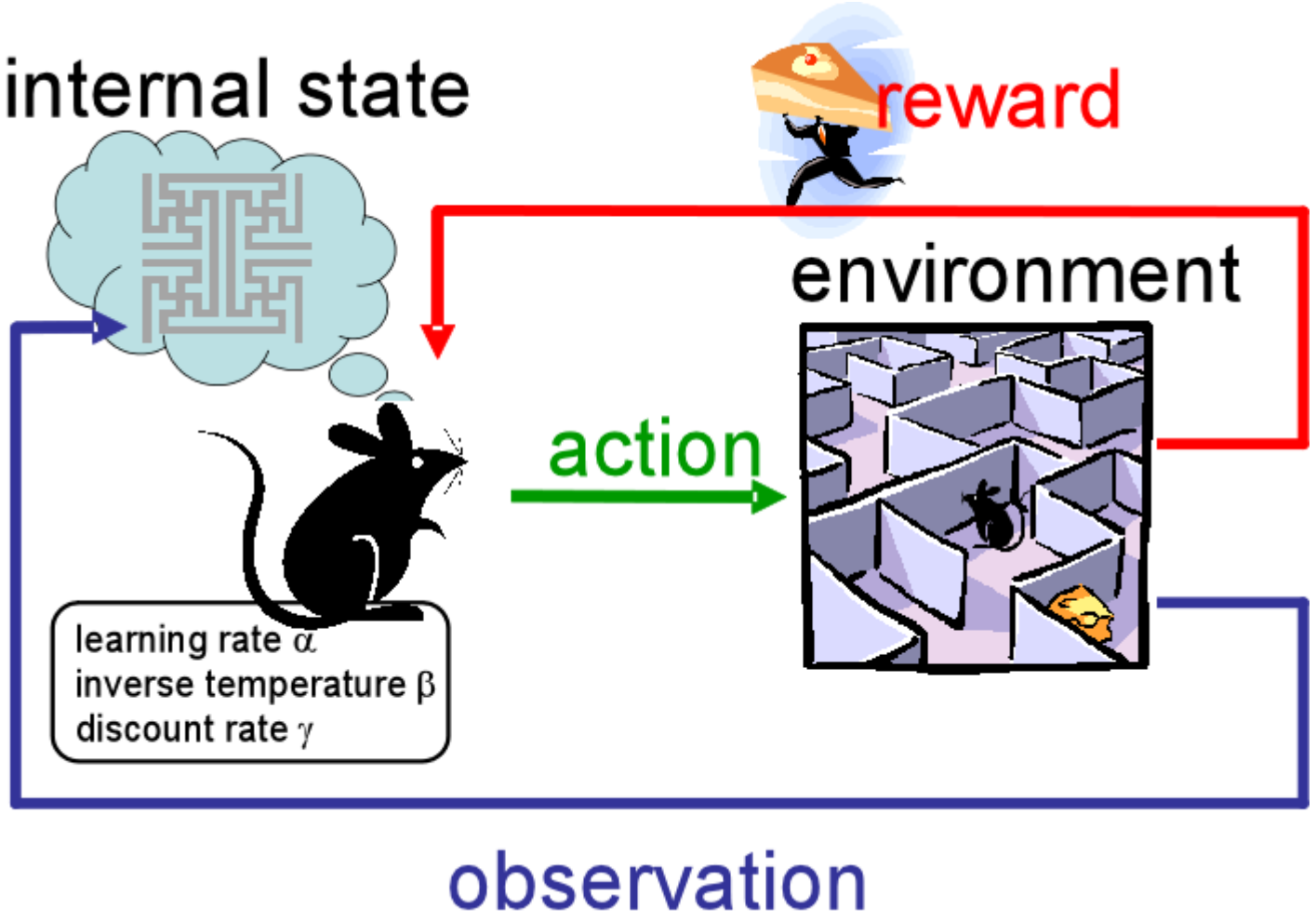
Instructor: Jivko Sinapov

Announcements

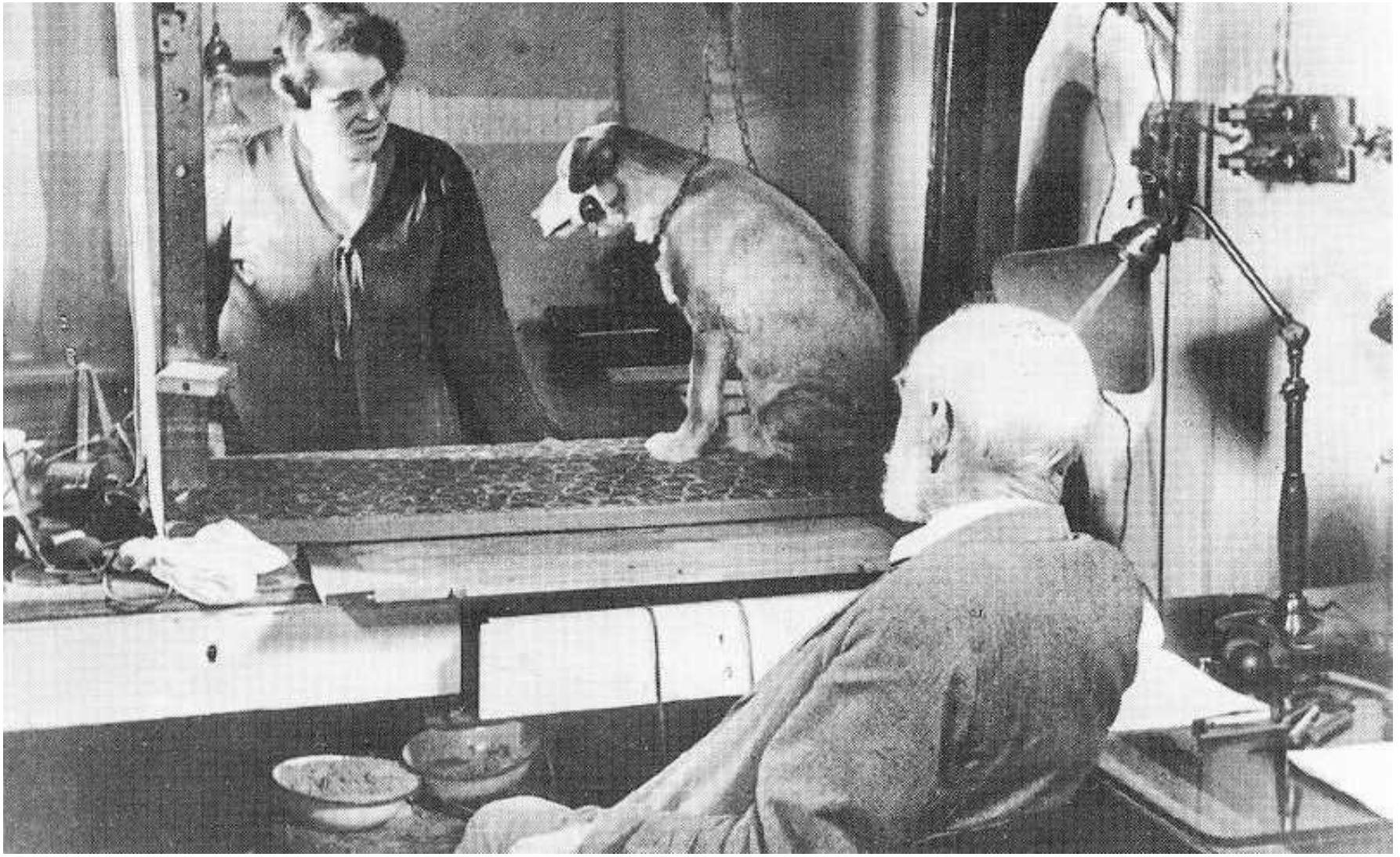
Reading Assignments

- Chapter 3 from Sutton and Barto
- Reading Responses due Tuesday (ideally before class)

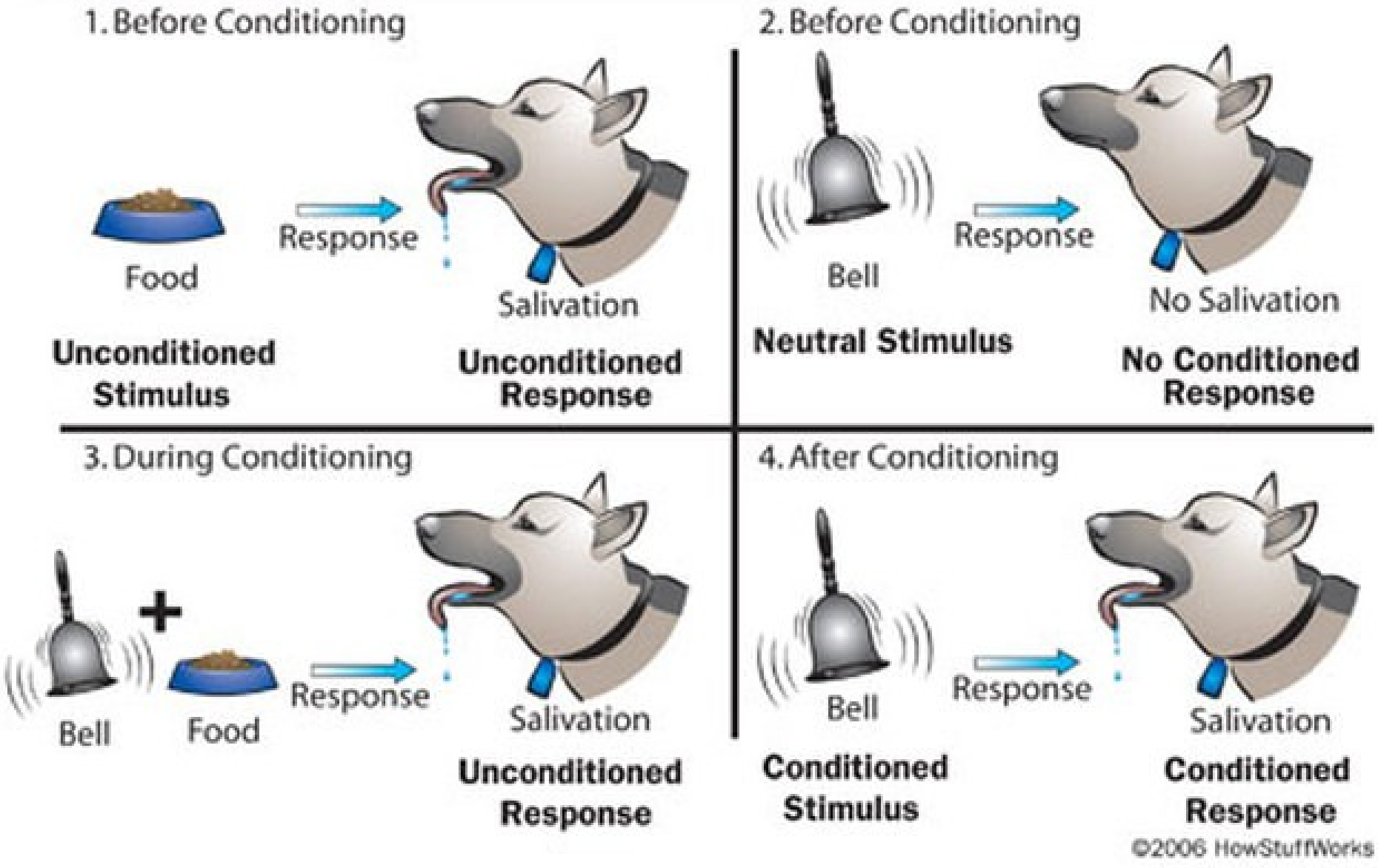
Reinforcement Learning

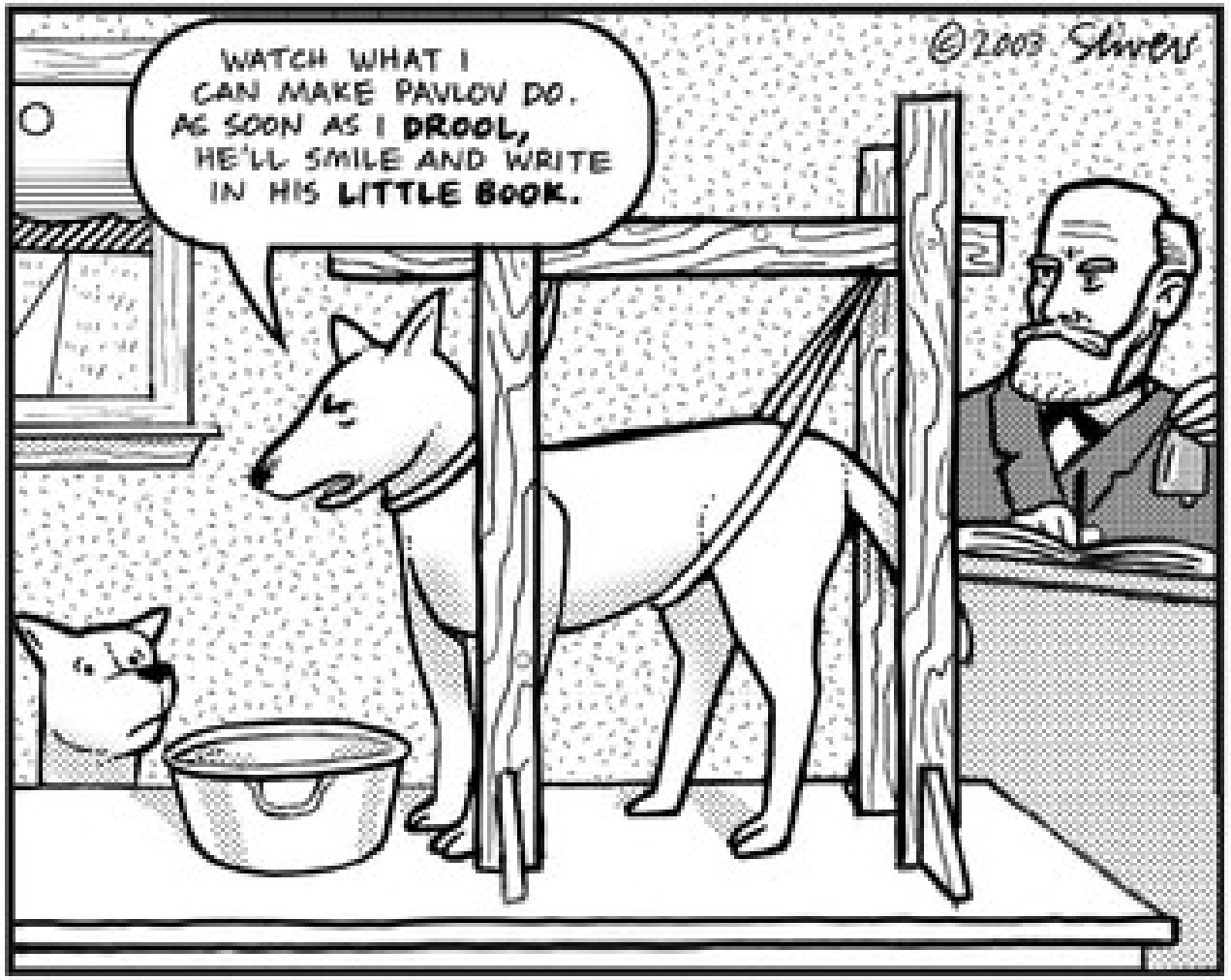


Ivan Pavlov (1849-1936)



How Dog Training Works

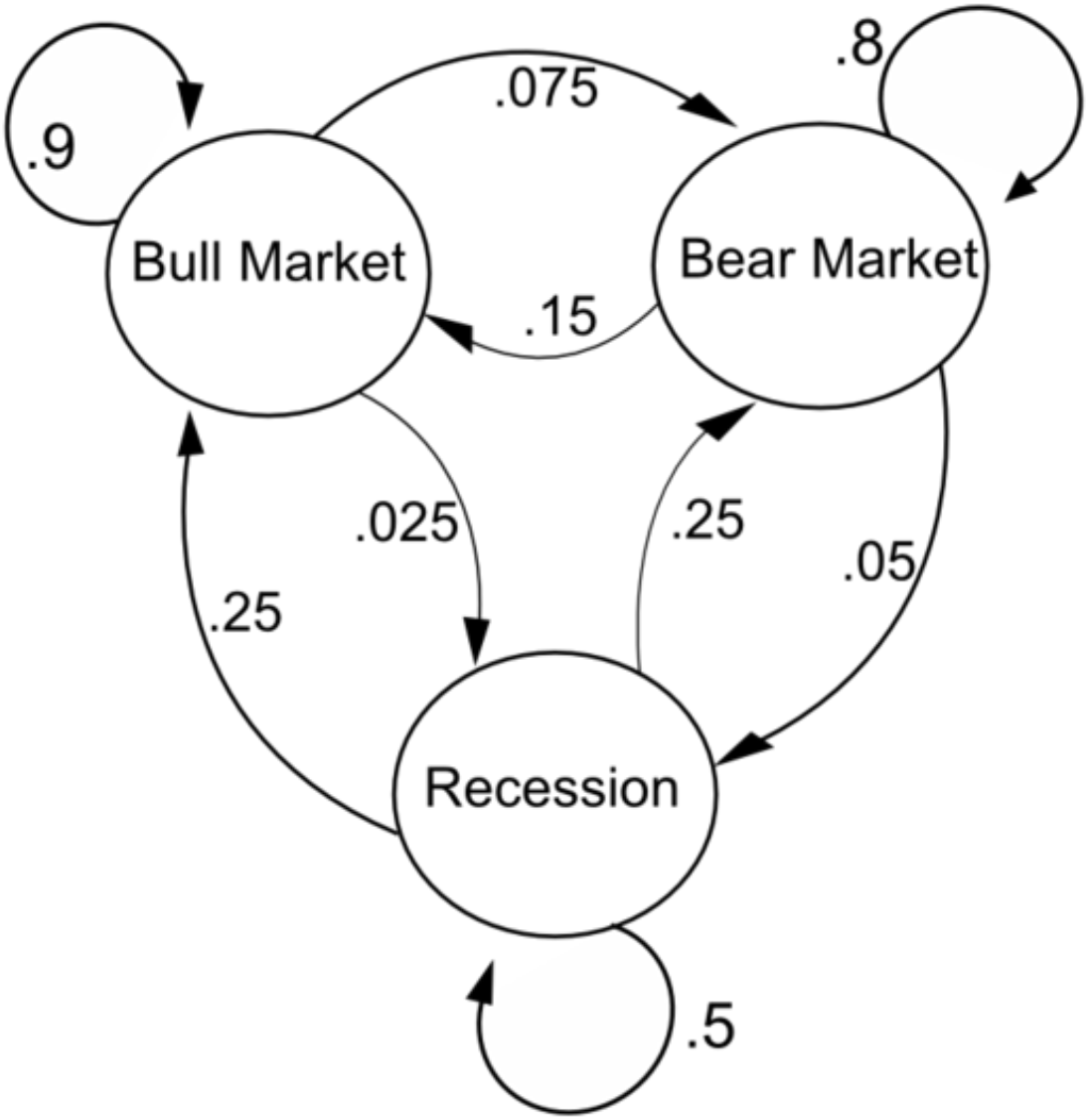



WATCH WHAT I CAN MAKE PAVLOV DO. AS SOON AS I DROOL, HE'LL SMILE AND WRITE IN HIS LITTLE BOOK.
©2003 Stiver

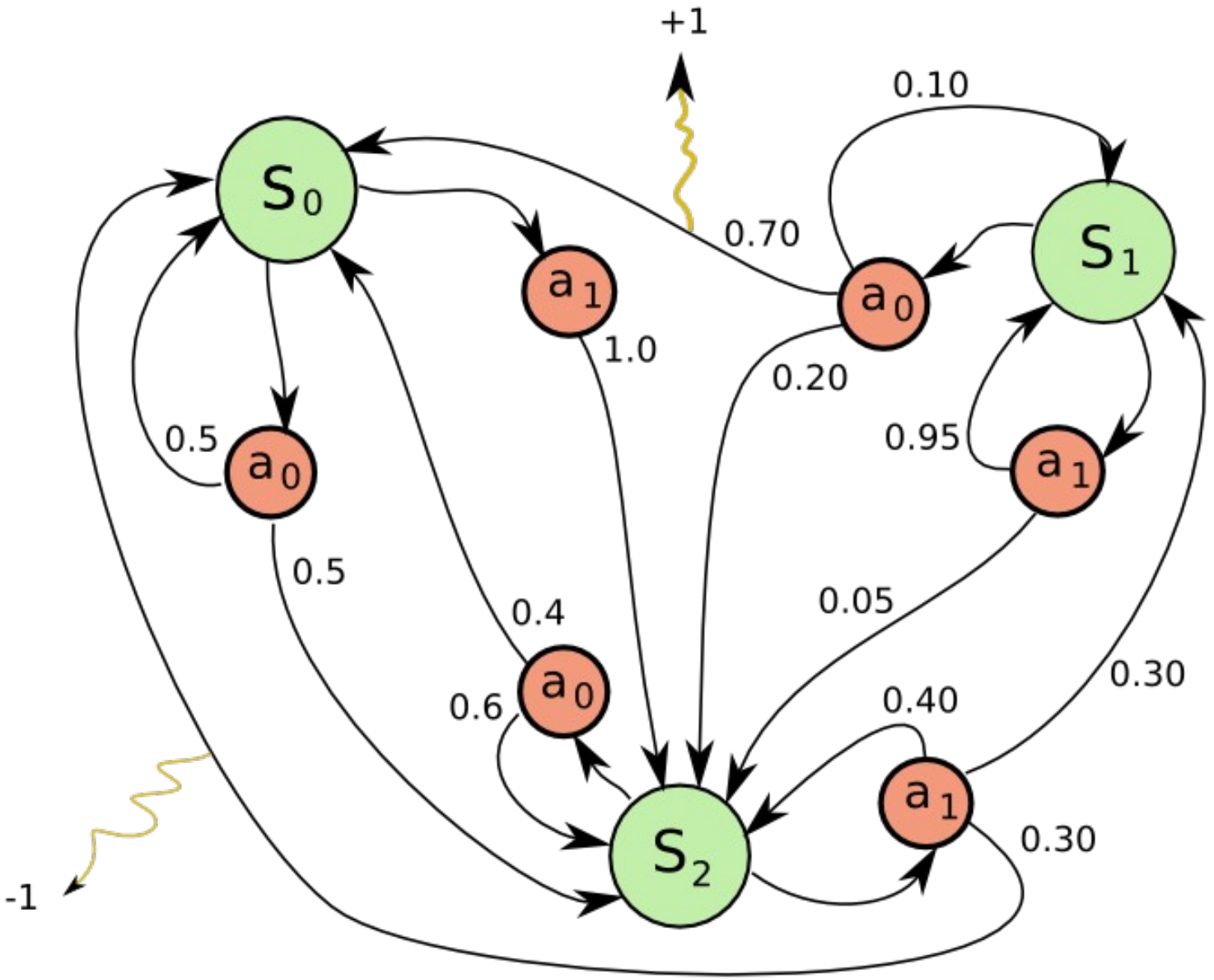
Andrey Andreyevich Markov (1856 – 1922)

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

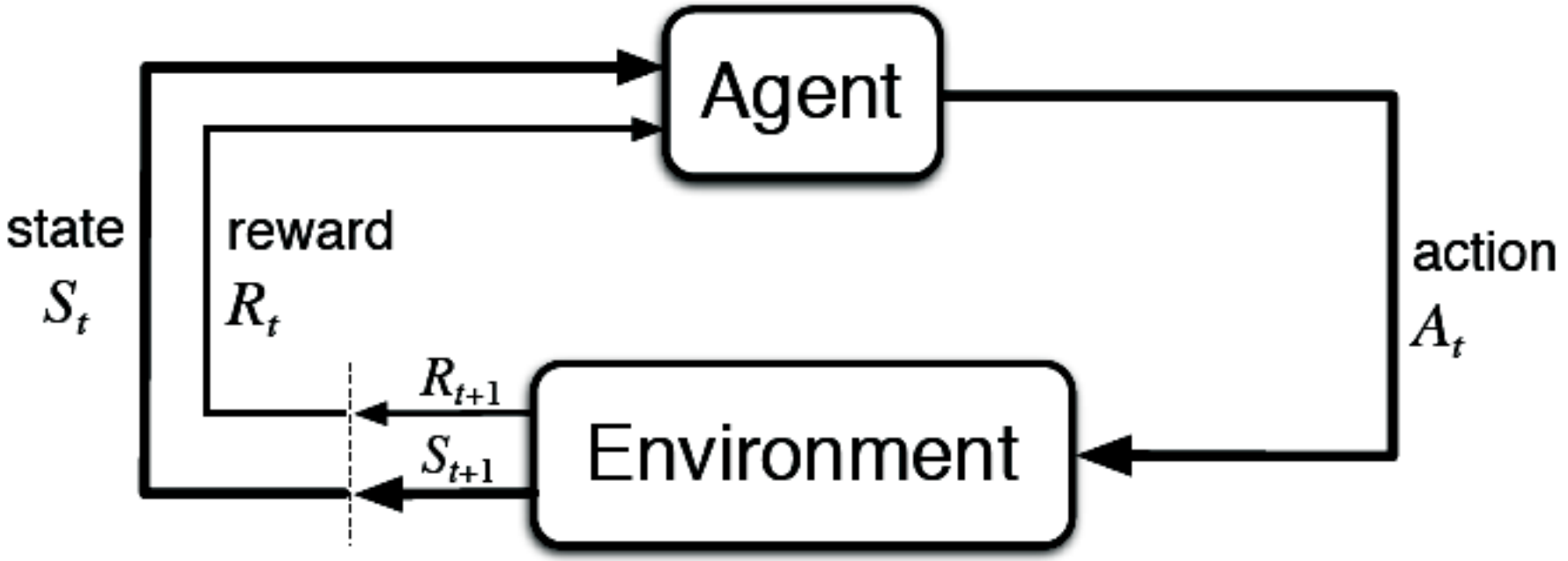
Markov Chain



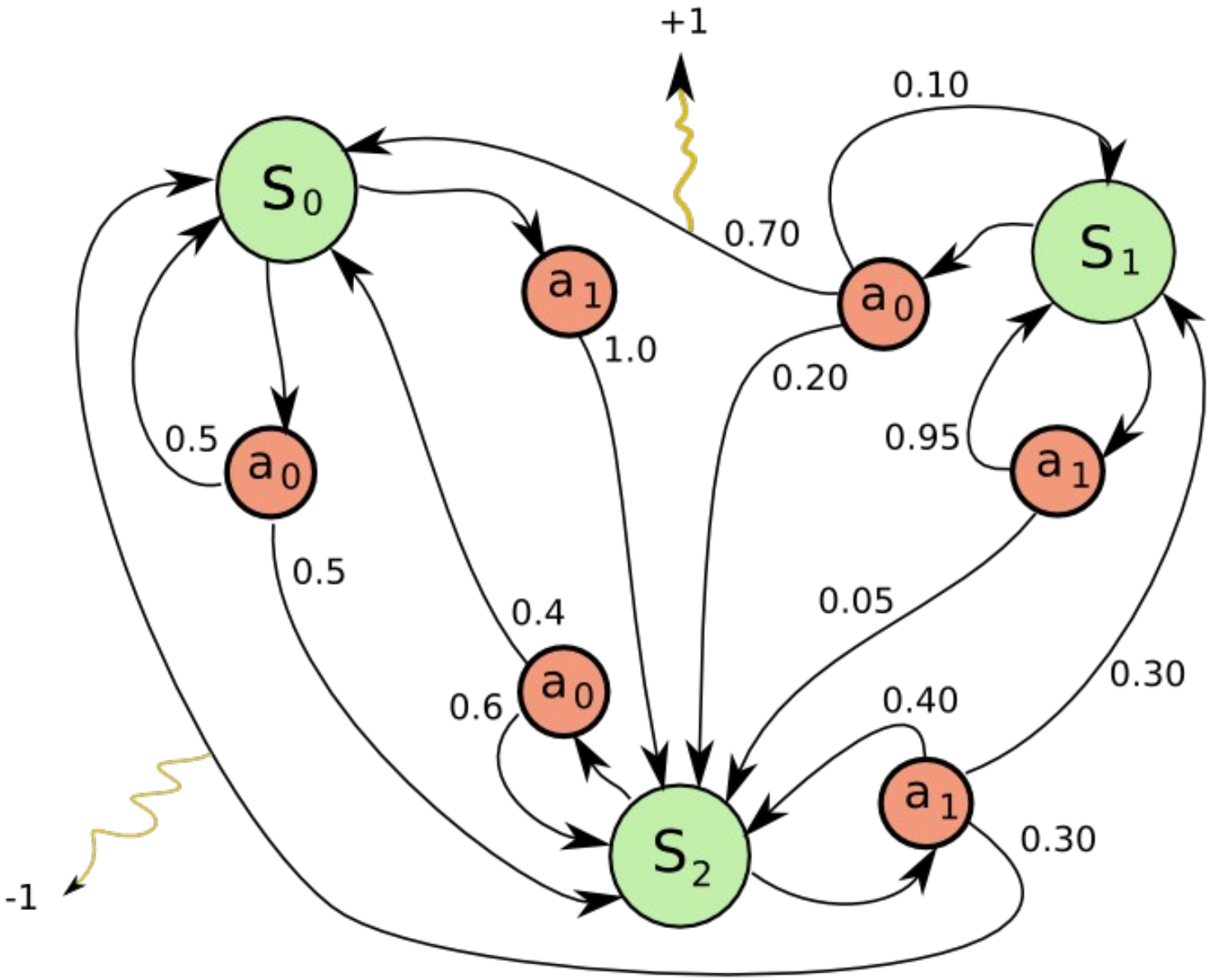
Markov Decision Process



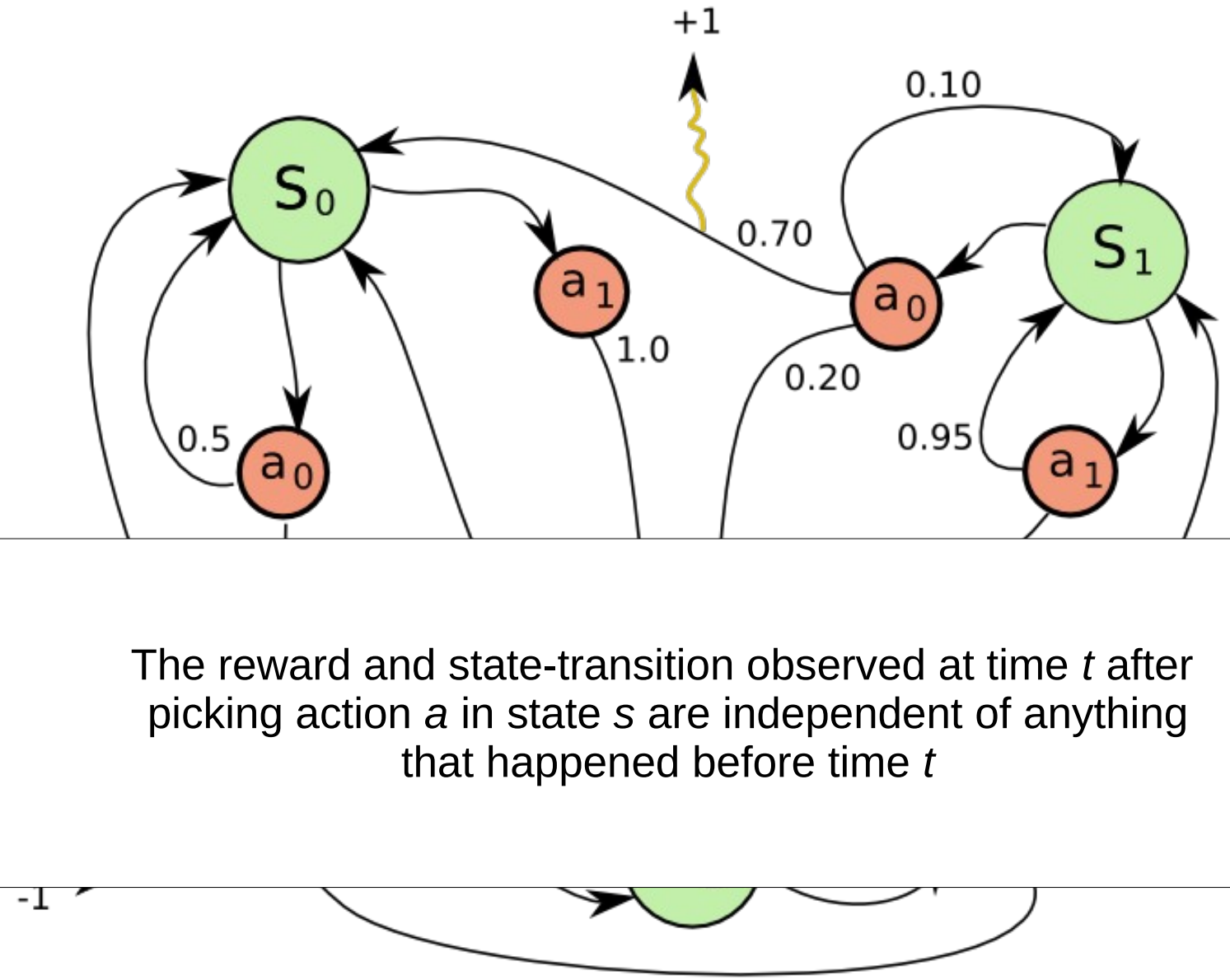
The Reinforcement Learning Problem



RL in the context of MDPs



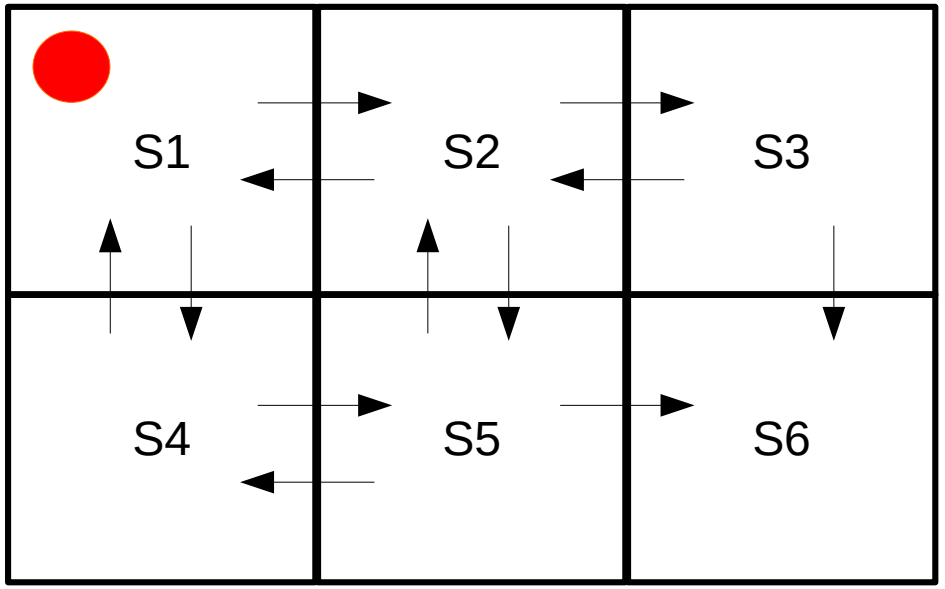
The Markov Assumption



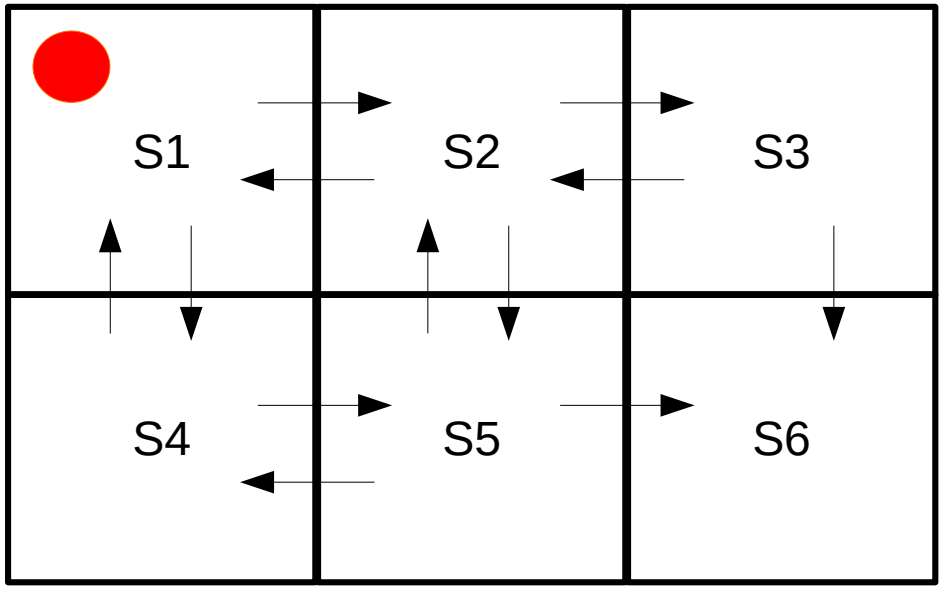
The reward and state-transition observed at time *t* after picking action *a* in state *s* are independent of anything that happened before time *t*

Q-Learning

(board exercise)



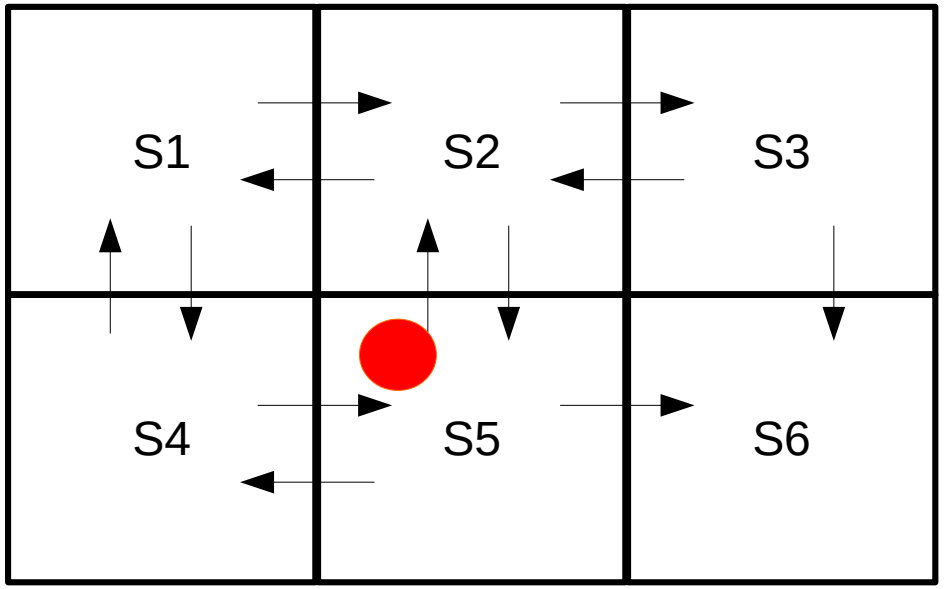
+ 100 reward for getting to S6
0 for all other transitions



+ 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

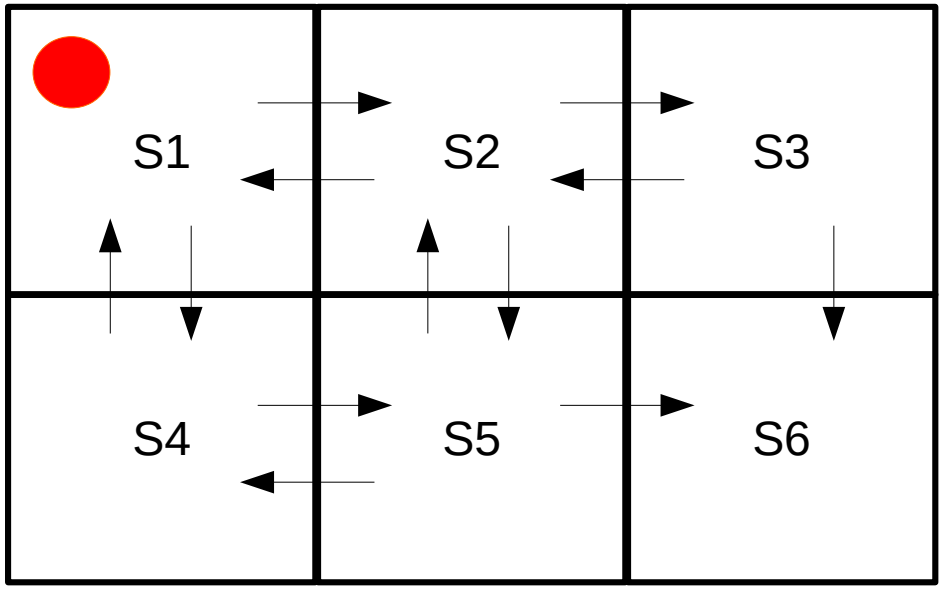
Update rule upon executing action a in state s, ending up in state s' and observing reward r :

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

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 :

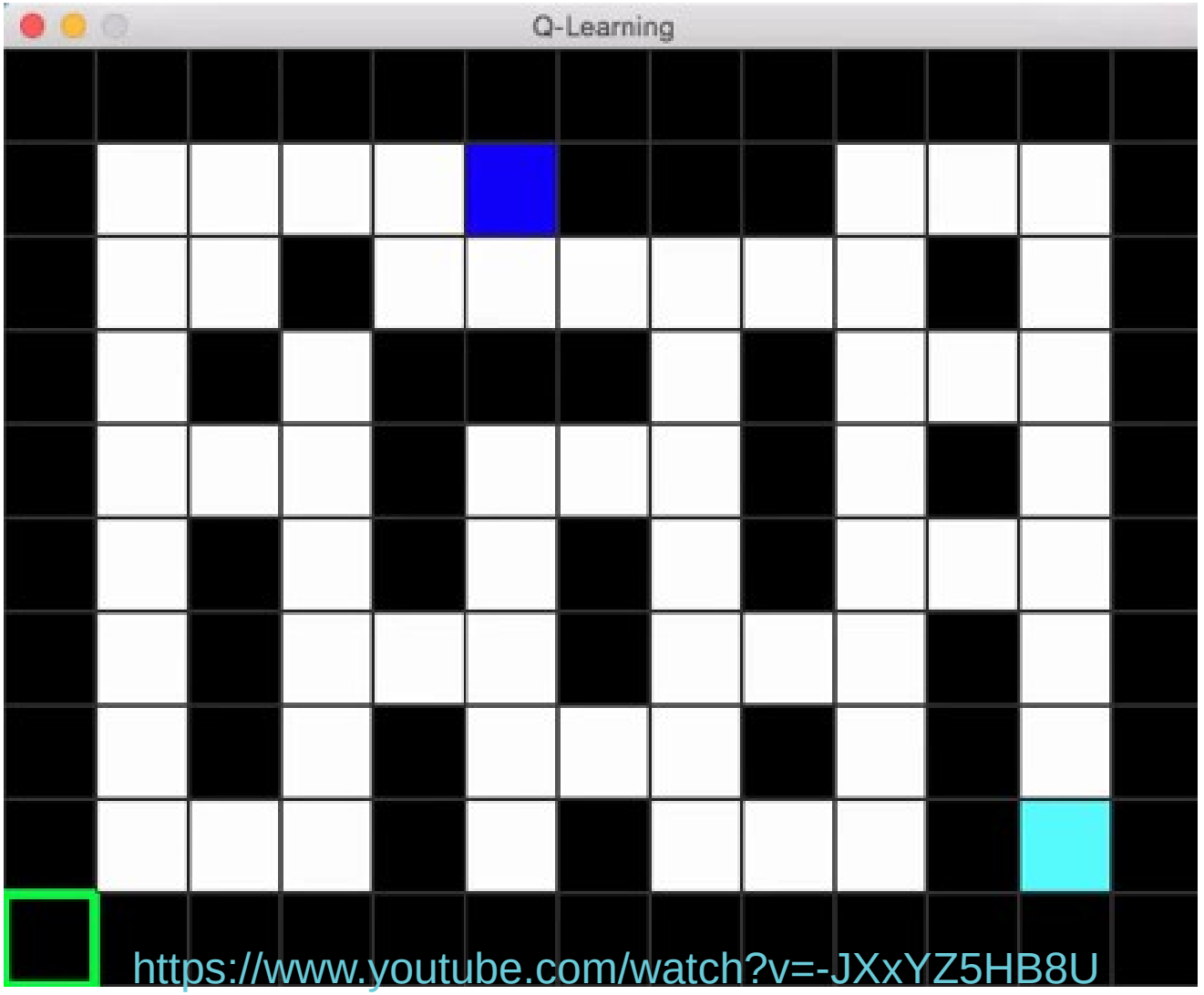
$$Q(s , a)=r +\gamma \max a' Q(s' ,a ')$$

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



Another Example

https://www.youtube.com/watch?v=zN34MieXjlU

Q-Learning Algorithm

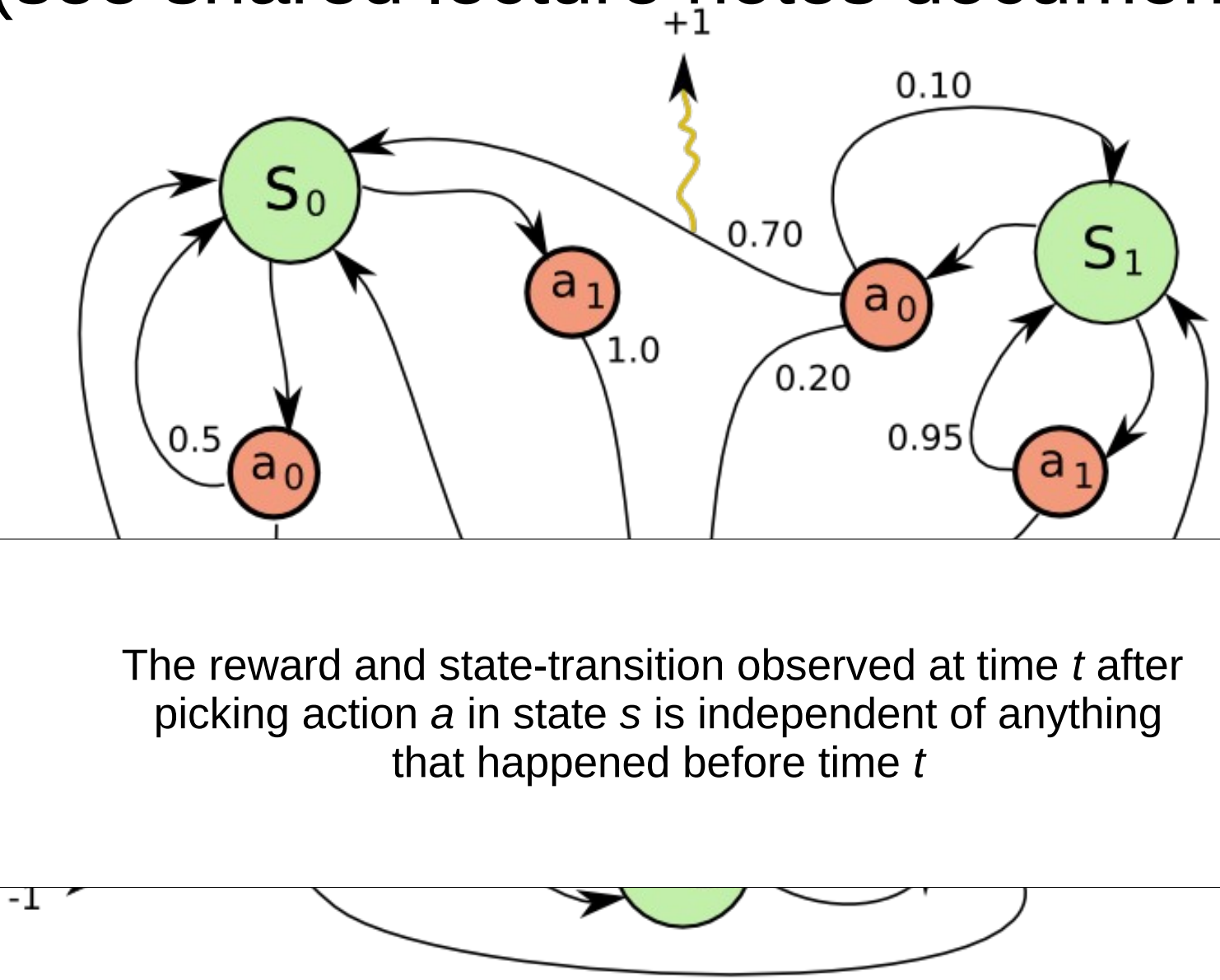
Q-learning (off-policy TD control) for estimating $\pi \approx \pi_*$

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

Example 3.3 and Exercise 3.4

The Agent-Environment Interface

The environment class

- Data: mean rewards for each arm, number of arms, variance of arms; other variables that dictate the non-stationary behavior of the bandit
- Functions:
  - pull_arm(int k)
  - change_bandit(….)
  - init / reset

The Agent Class

- Data:
-
-
-
- Functions