



LUND
UNIVERSITY

agent



INTRODUCTION TO REINFORCEMENT LEARNING

Volker Krueger

with many slides stolen from David Silver's lecture

- I. Markov Decision Processes:
 - I. The basics
- 2. Iterative Policy Evaluation:
 - 1. The first steps,
 - 2. get better understanding
- 3. Monte Carlo Prediction and Control
 - 1. the first real RL approach
 - 2. working OK
- 4. (Temporal Difference Learning, $\text{TD}(\lambda)$)
 - I. Works much better than MC Prediction and Control)



LUND
UNIVERSITY

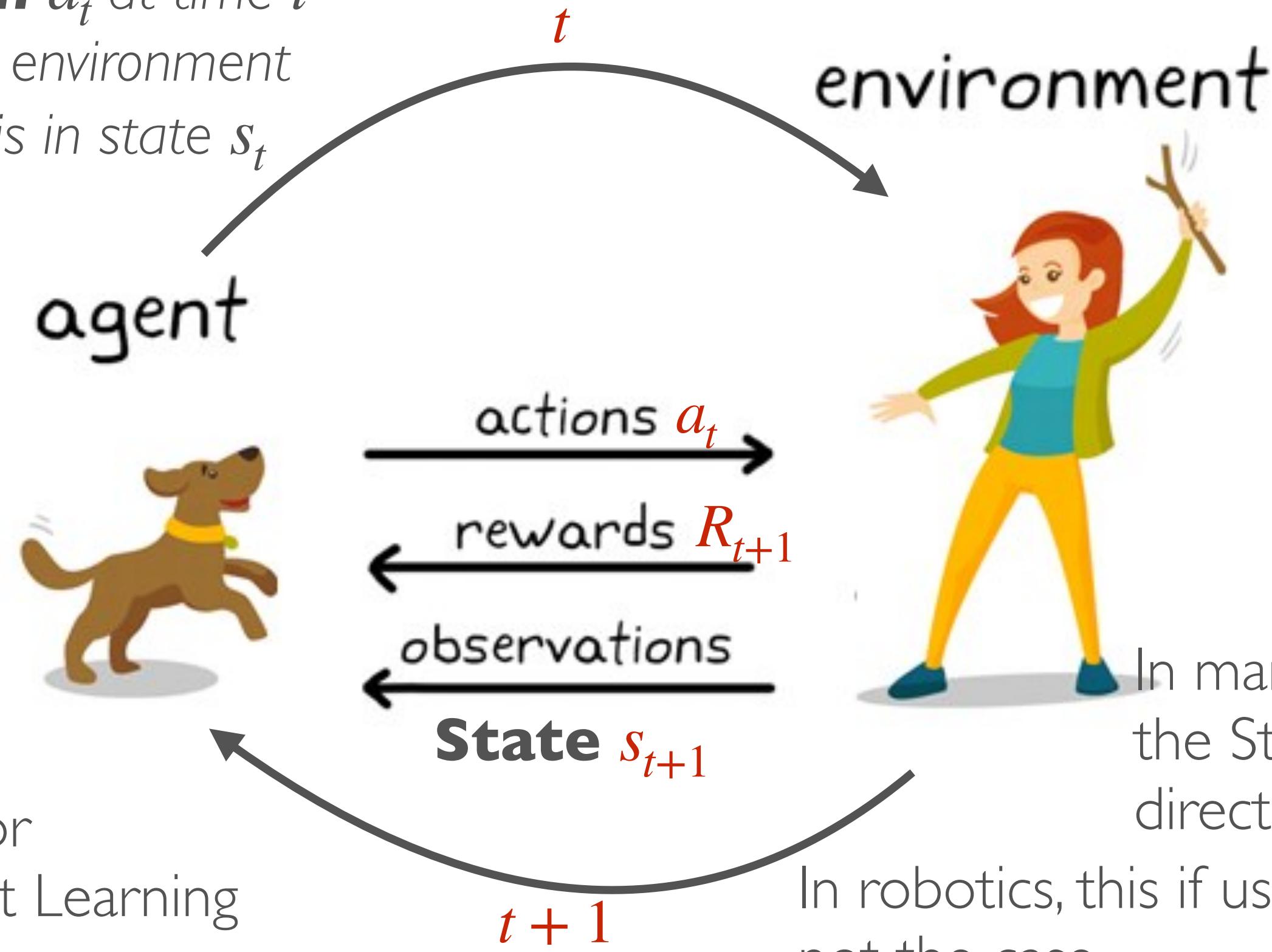
MARKOV DECISION PROCESSES

Our Task for Today

Slides partially borrowed from a presentation by David Silver



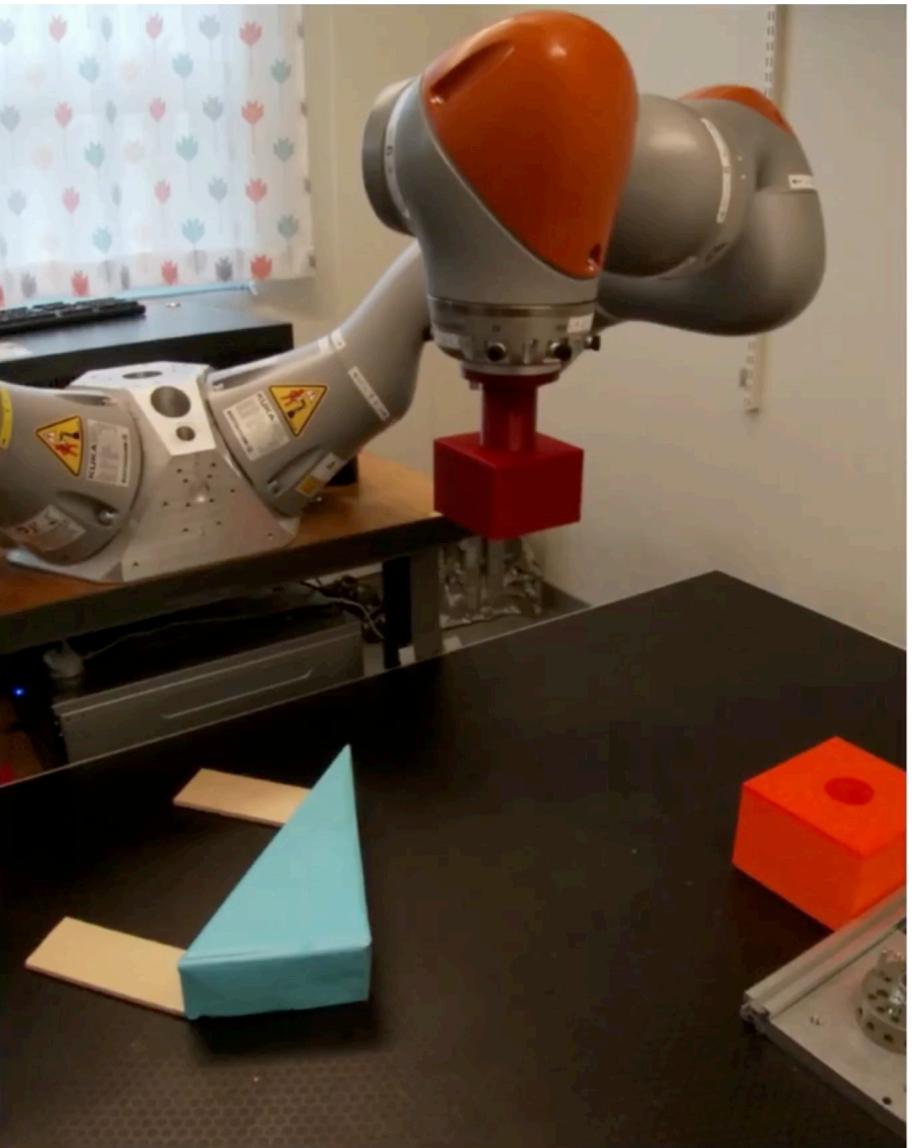
Applies an
Action a_t at time t
to the environment
that is in state s_t



Model used for
Reinforcement Learning

In many cases
the State can be
directly observed
In robotics, this is usually
not the case

REINFORCEMENT LEARNING



Throwing tests
with varying attitudes.

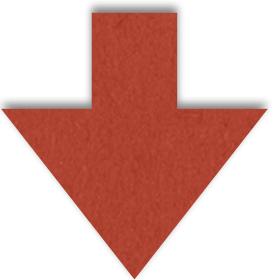
- Value function + Policy modelled using CNNs.
- 14 robots needed 3000h (800.000 grasp attempts) before something intelligent happened

EXAMPLE: PICKING MARKOV CHAIN



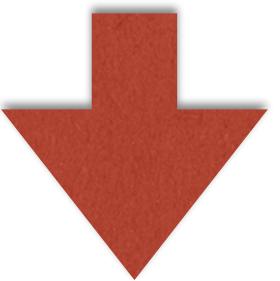
OVERVIEW FOR TODAY

Markov Processes (MP)



+ Rewards

Markov Reward Processes (MRP)



+ Actions

Markov Decision Processes (MDP)

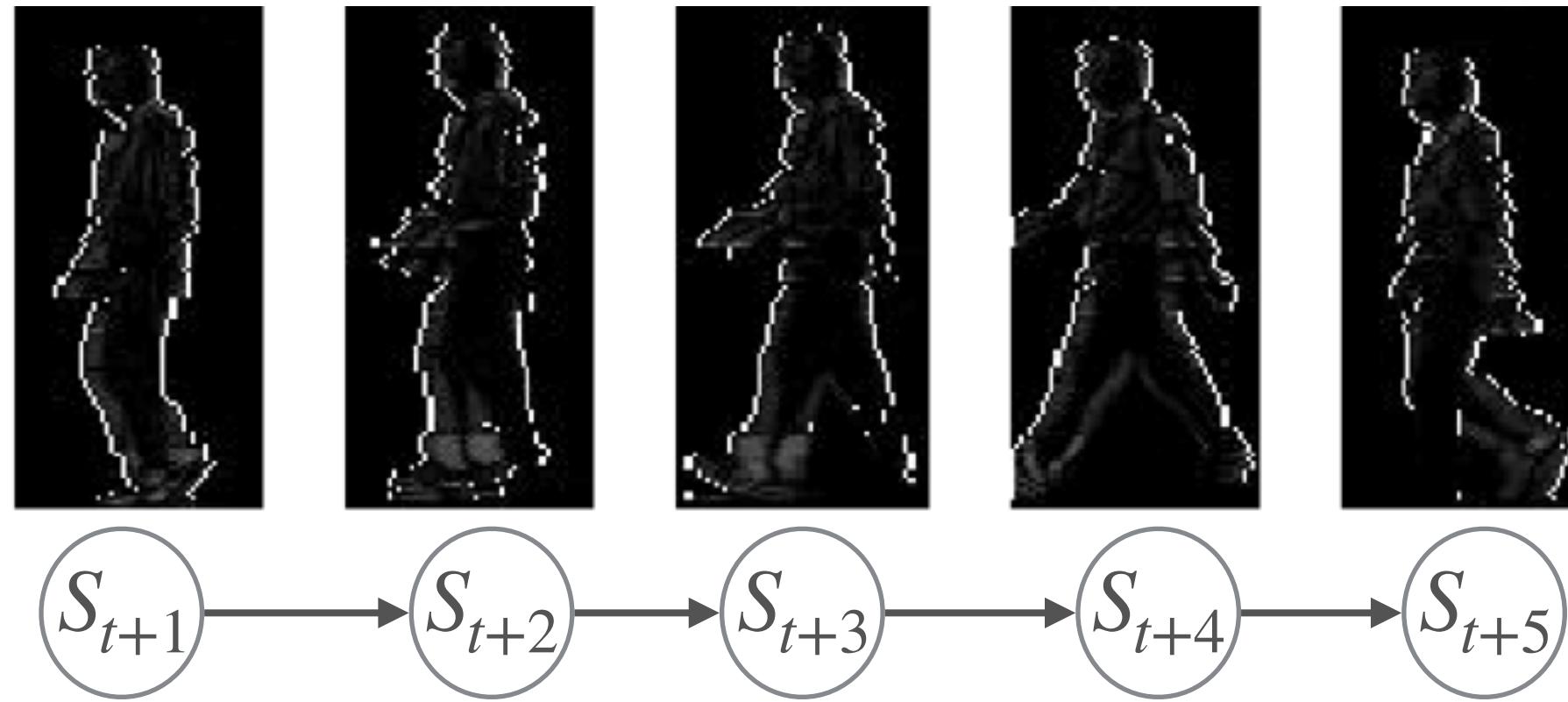
- Formal Model of Reinforcement Learning Tasks
- Value Functions
- Bellman Equation
- Optimal Value Function: *solving the MDP*



MARKOV PROCESS



MARKOV PROCESSES



- The next state is dependent **only** on the present state
- A random state S_t is Markov if and only if

$$P(S_{t+1} | S_t) = P(S_{t+1} | S_1, \dots, S_t)$$

STATE TRANSITION MATRIX

- For a Markov state s and a successor state s' , the *state transition probability* is defined by

$$P_{ss'} = P(S_{t+1} = s' | S_t = s)$$

- A state transition probability matrix summarises the possible state transitions

$$\mathbf{P} = \begin{bmatrix} P_{11} & \dots & P_{1n} \\ \vdots & & \\ P_{n1} & \dots & P_{nn} \end{bmatrix}$$

Note that each row of \mathbf{P} sums to 1.



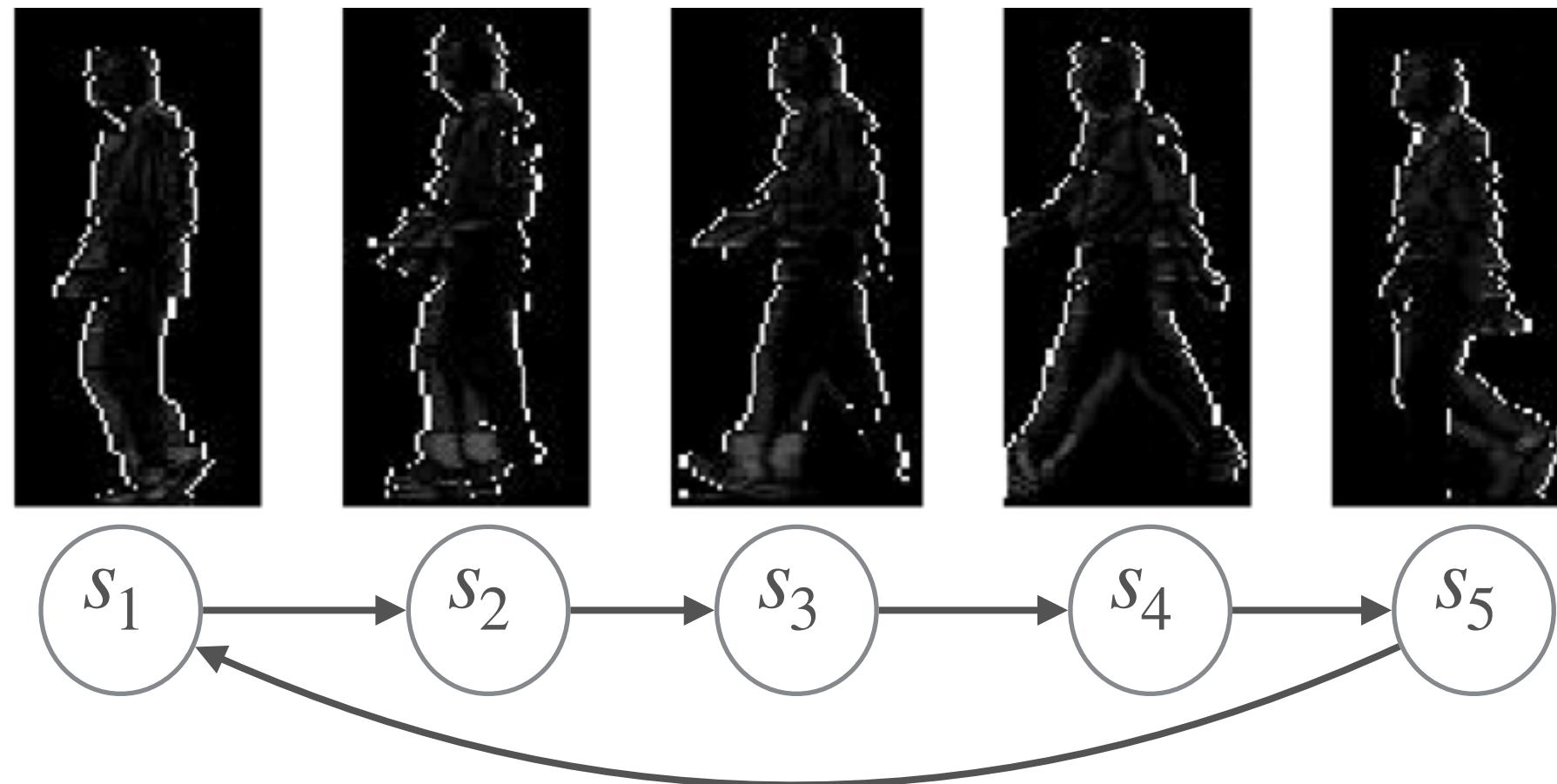
MARKOV PROCESS

- A Markov process is a sequence $X_t = s, X_{t+1} = s', \dots$ of Markov random variables, i.e., a sequence of random variables that have the Markov property
- **Definition:** A Markov process (or Markov Chain) is a tuple (\mathbf{S}, \mathbf{P})
 - $\mathbf{S} = \{s_1, s_2, \dots, s_N\}$ is a (finite) set of states
 - \mathbf{P} is a state transition probability matrix, with

$$P_{ss'} = P(X_{t+1} = s' | X_t = s)$$



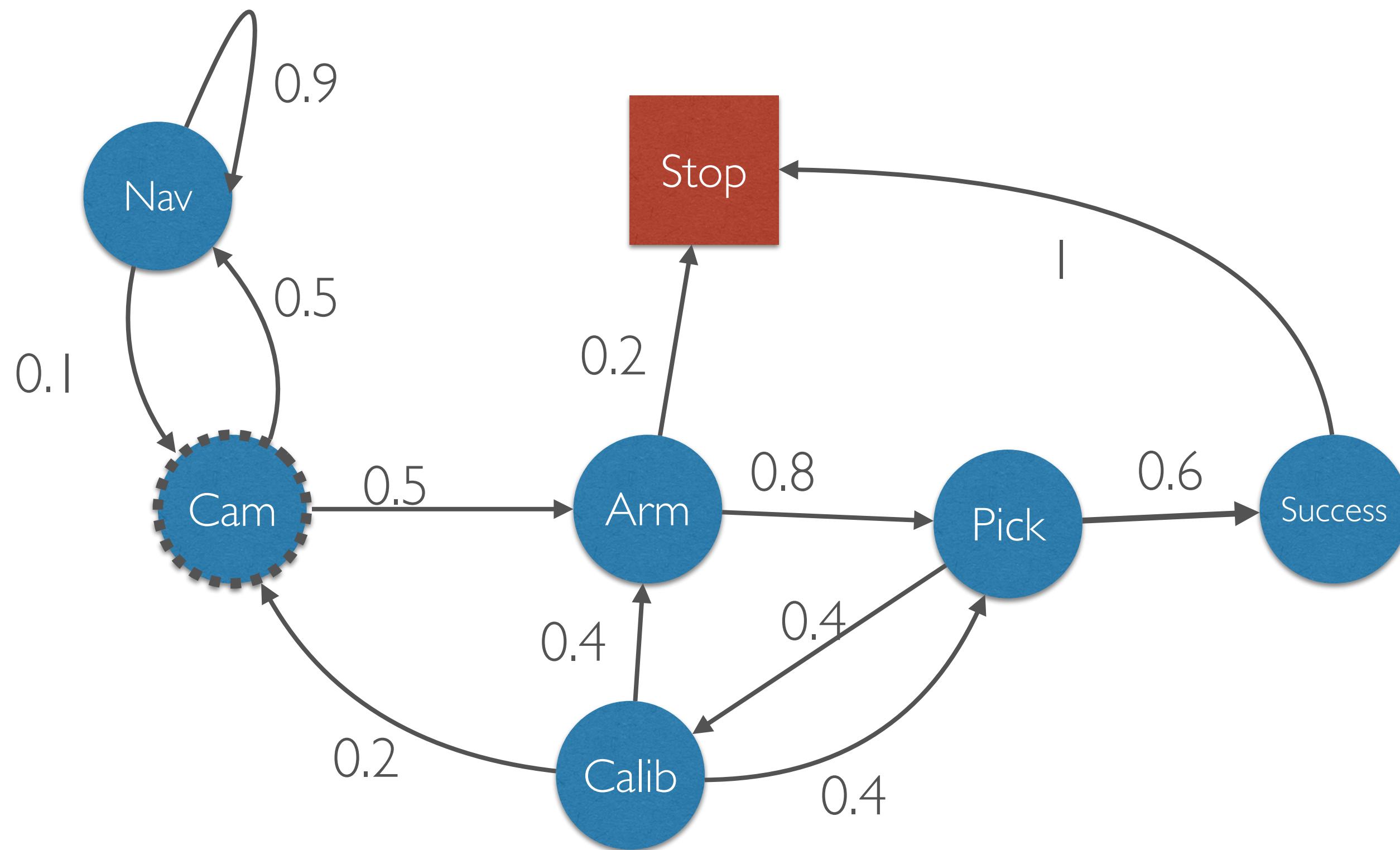
EXAMPLE GAIT MARKOV CHAIN



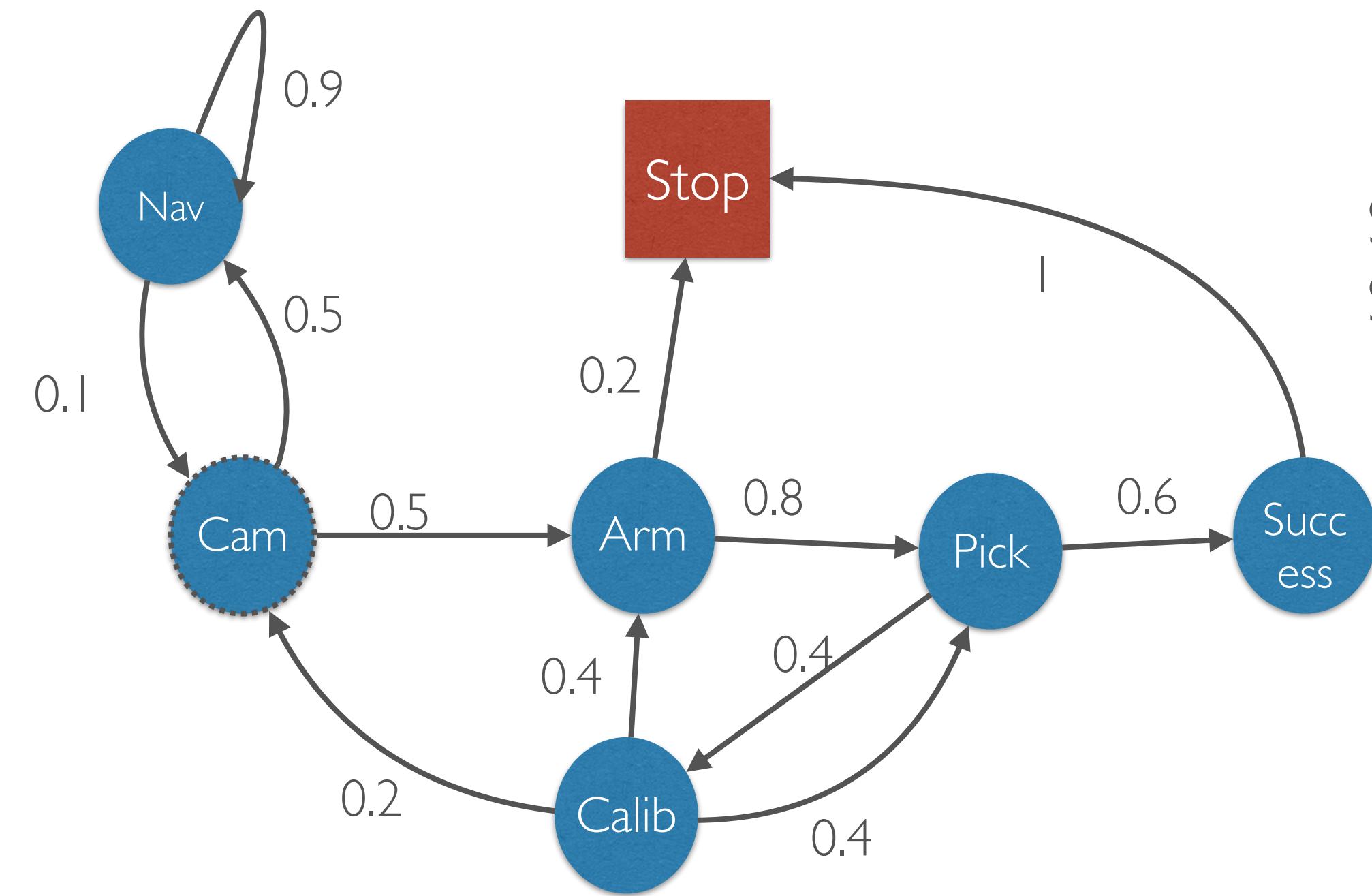
What are the
transition
probabilities?



EXAMPLE: PICKING MARKOV CHAIN



EXAMPLE: PICKING MARKOV CHAIN



Sample **episodes**:

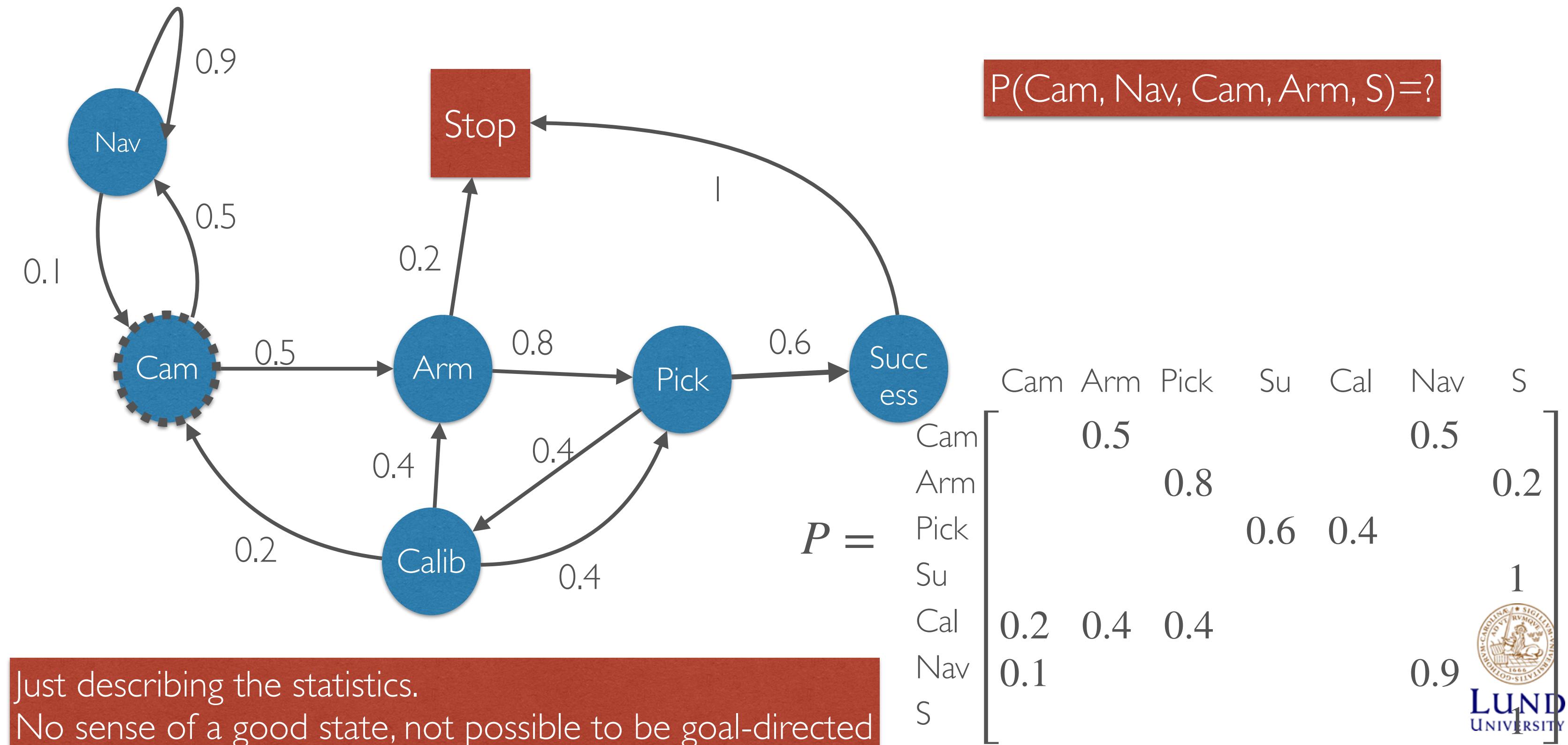
Start state $S_1 = \text{Cam}$

S_1, S_2, \dots, S_T

- Cam, Nav, Cam, Arm, Pick, Su, S
- Cam, Arm, Pick, Calib, Pick, Su, S
- Cam, Nav, Cam, Arm, S



EXAMPLE: PICKING MARKOV CHAIN



MARKOV REWARD PROCESSES



LUND
UNIVERSITY

MARKOV REWARD PROCESS

- **Definition:** A Markov Reward Process is a tupel $(\mathbf{S}, \mathbf{P}, \mathbf{R}, \gamma)$
 - \mathbf{S} is a finite set of states
 - \mathbf{P} is a state transition probability matrix $P_{ss'} = P(S_{t+1} = s' | S_t = s)$

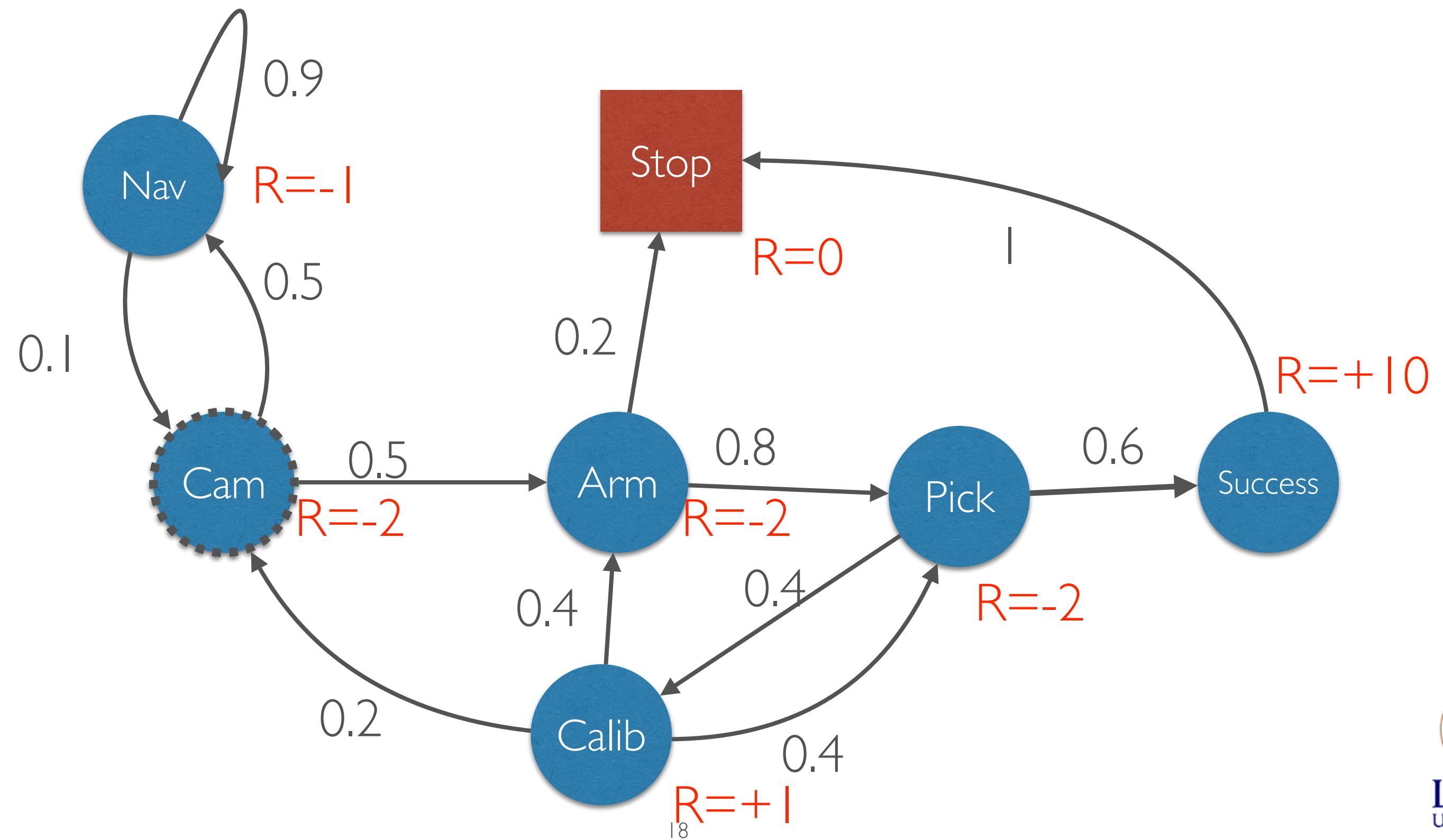
- \mathbf{R} is a reward function, $R_s = E\{R_{t+1} | S_t = s\}$
- γ is a discount factor, $\gamma \in [0,1]$

get a reward for being in
a certain state

*Being in a certain state
can be very rewarding*



EXAMPLE: PICKING MARKOV CHAIN



RETURN

Definition: The **return** G_t is the total discounted reward starting from time-step t .

$$G_t = R_{t+1} + \gamma R_{t+2} + \dots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

- The **discount factor** $\gamma \in [0,1]$ weights the future rewards.
- The value of a reward R that is $k+1$ steps into the future is $\gamma^k R$
 - Short-sighted: γ close to 0
 - Far-sighted: γ close to 1

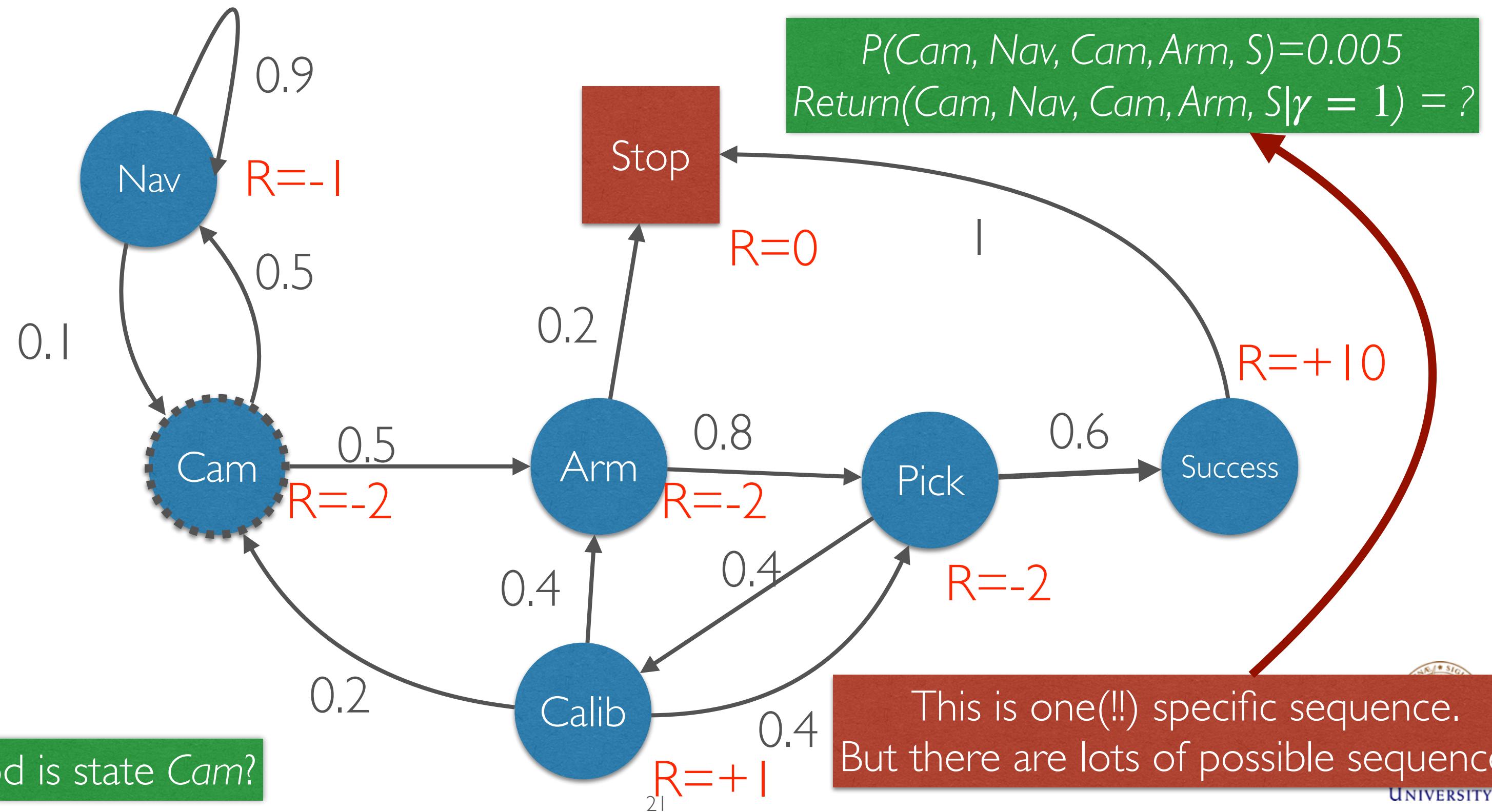
WHY DO WE WANT γ ?

$$G_t = R_{t+1} + \gamma R_{t+2} + \dots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

- Mathematical convenience as γ^k vanishes with k for $\gamma < 1$
 - Assures finite returns even in cyclic Markov processes
- Represents the uncertainty about the future rewards
- Animal/human behaviour shows preference for immediate reward
- Possible to use *undiscounted* Markov reward process ($\gamma = 1$) if all sequences terminate (assures finite return)



EXAMPLE: PICKING MARKOV CHAIN



*Cam, Nav, Cam, Arm, Pick, Su, S
Cam, Arm, Pick, Calib, Pick, Su, S
Cam, Nav, Cam, Arm, S*

How good is it to be in, .e.g, the state *Cam*?

What is the expected return considering all possible sequences?



$$\gamma = 0 ?$$

$$\gamma = 0.9 ?$$



STATE VALUE FUNCTION

- The state value function $v(s)$ gives the *long-term value* of a state s

Definition: The **state value function** $v(s)$ of a MRP is the expected return starting from state s

$$v(s) = E \{ G_t | S_t = s \}$$



RETURNS OF THE PICKING NETWORK

Sample **returns**: $S_1 = \text{Cam}$ with $\gamma = \frac{1}{2}$

$$G_1 = R_2 + \gamma R_3 + \dots + \gamma^{T-2} R_T$$

Sample **episodes**:

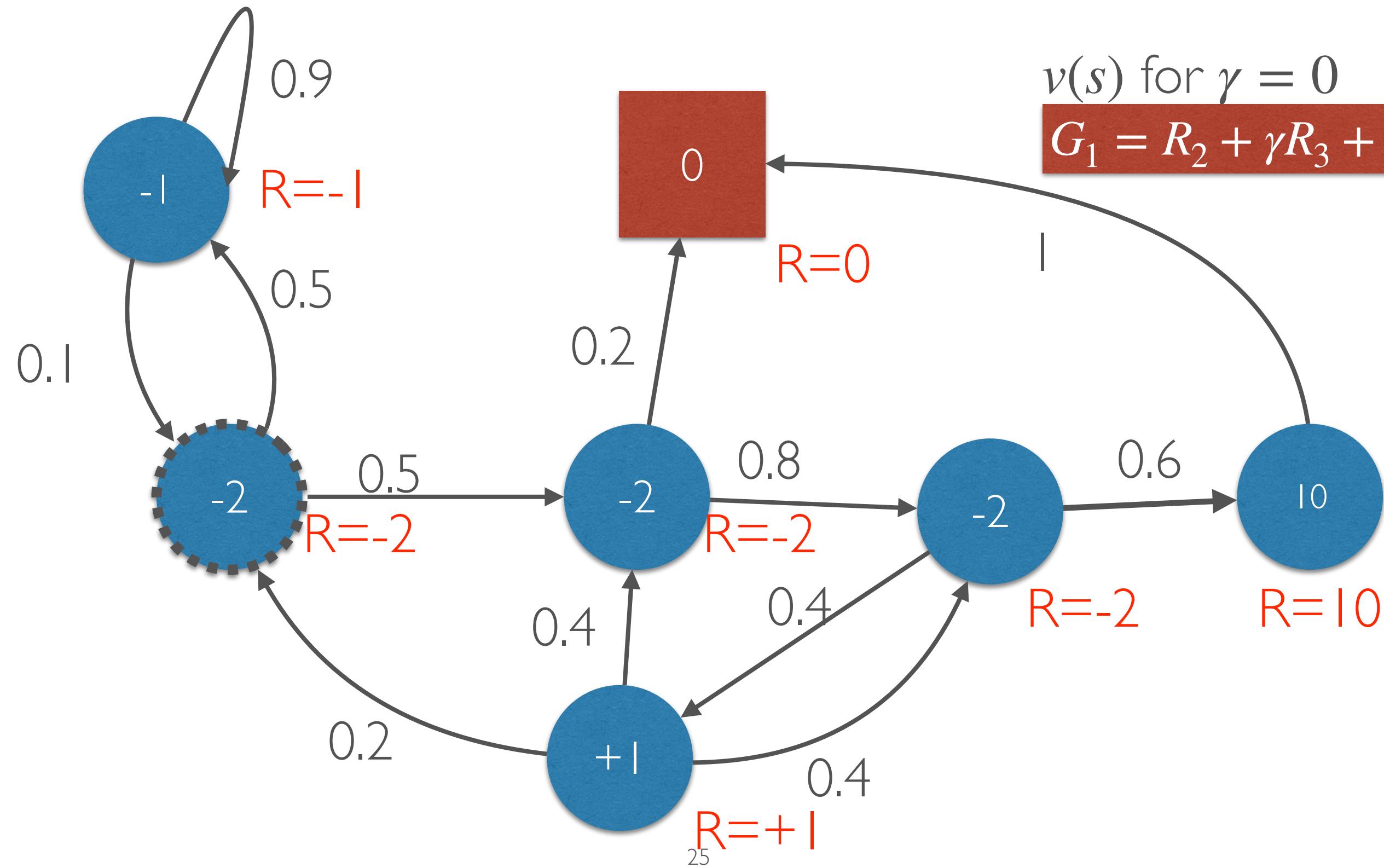
- Cam, Nav, Cam, Arm, Pick, Su, S
- Cam, Arm, Pick, Cal, Pick, Su, S
- Can, Nav, Cam, Arm, S

$$v_1 = -2 + \frac{1}{2}(-1) + \frac{1}{4}(-2) + \frac{1}{8}(-2) + \frac{1}{16}(-2) + \frac{1}{32}10 = -3.28$$

$$v_1 = -2 + \frac{1}{2}(-2) + \frac{1}{4}(-2) + \frac{1}{8}(1) + \frac{1}{16}(-2) + \frac{1}{32}10 = -3.18$$

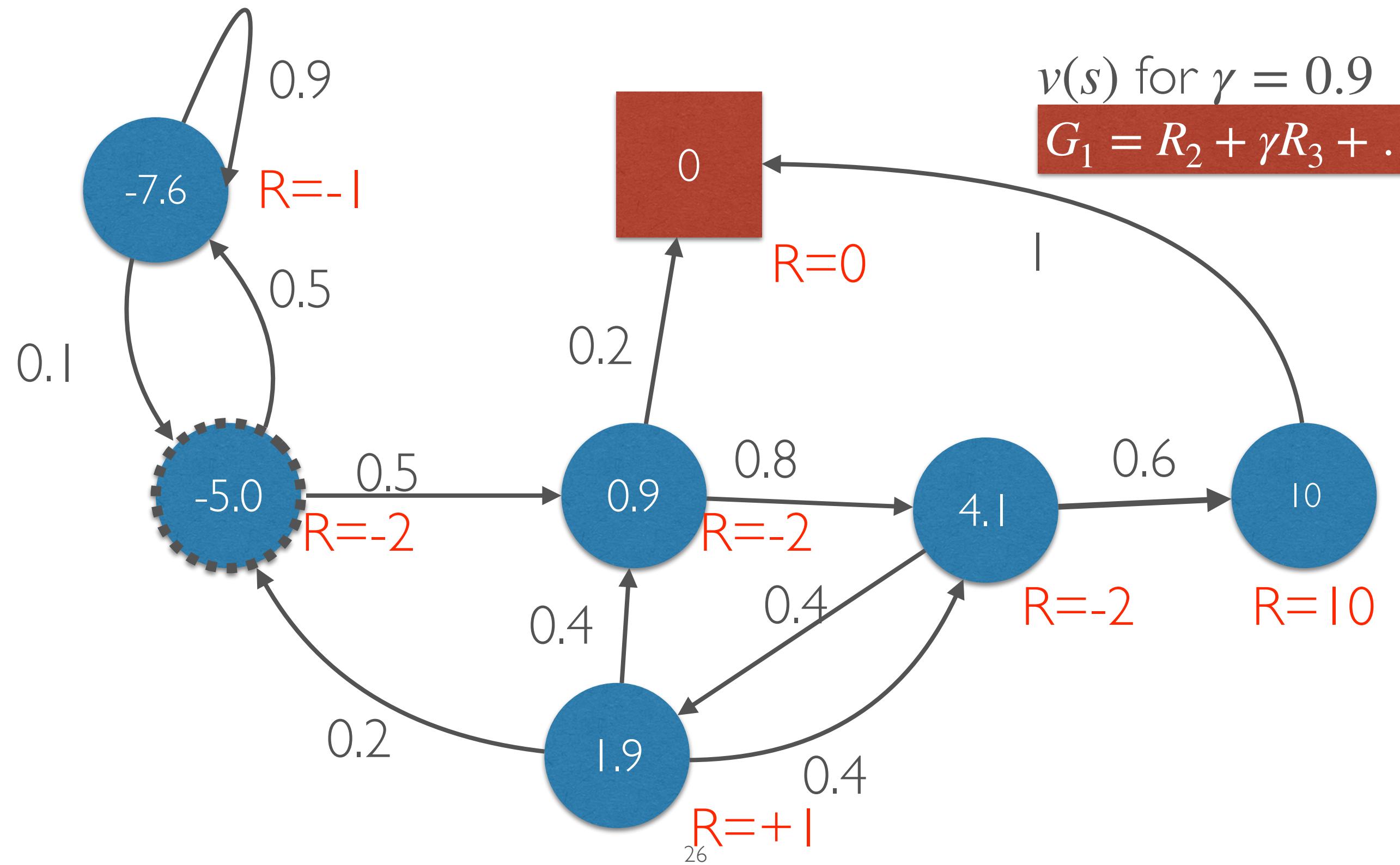
$$v_1 = -2 + \frac{1}{2}(-2) + \frac{1}{4}(-1) + \frac{1}{8}(-2) + \frac{1}{16}(-2) = -3.62$$

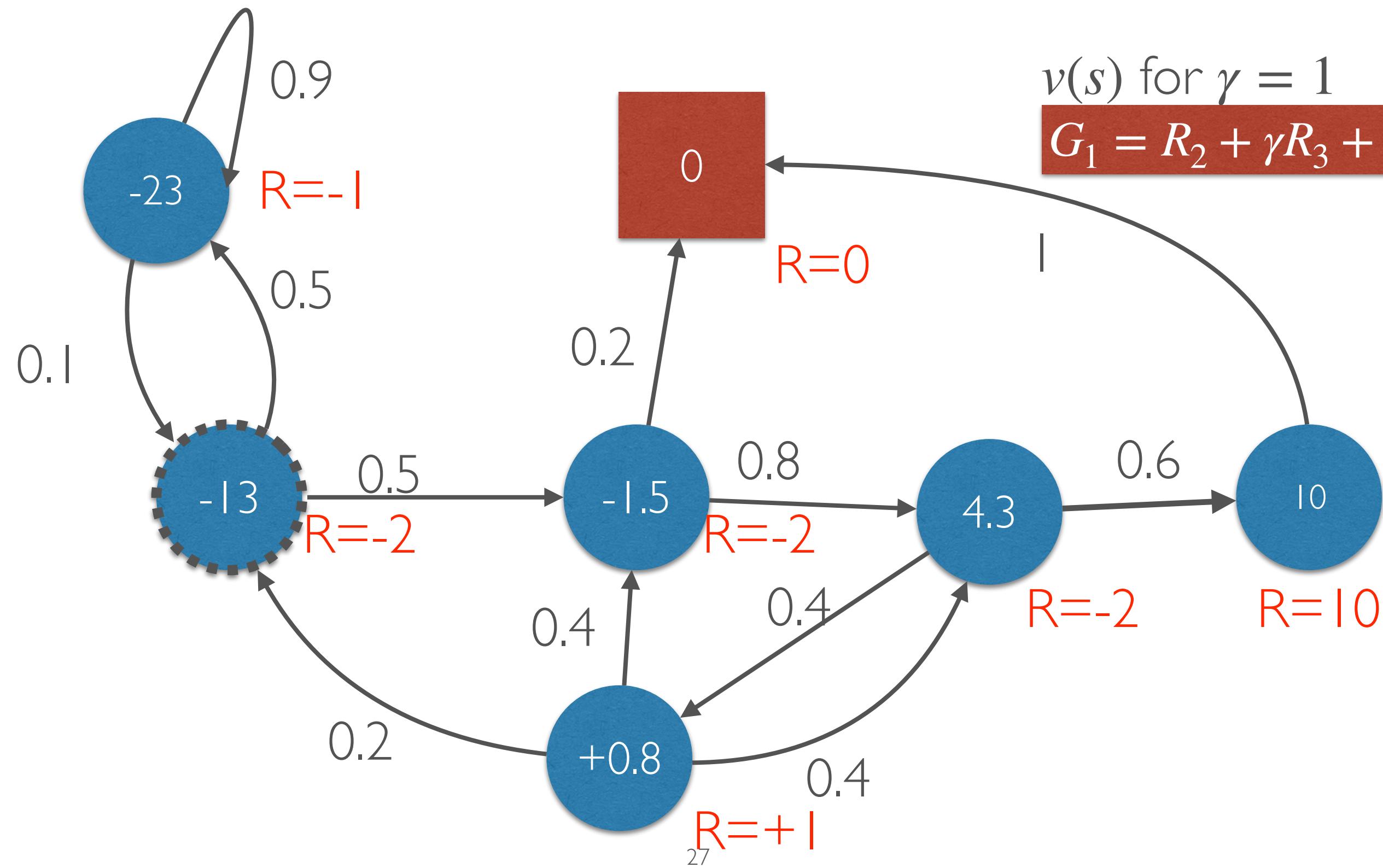




$v(s)$ for $\gamma = 0$

$$G_1 = R_2 + \gamma R_3 + \dots + \gamma^{T-2} R_T$$





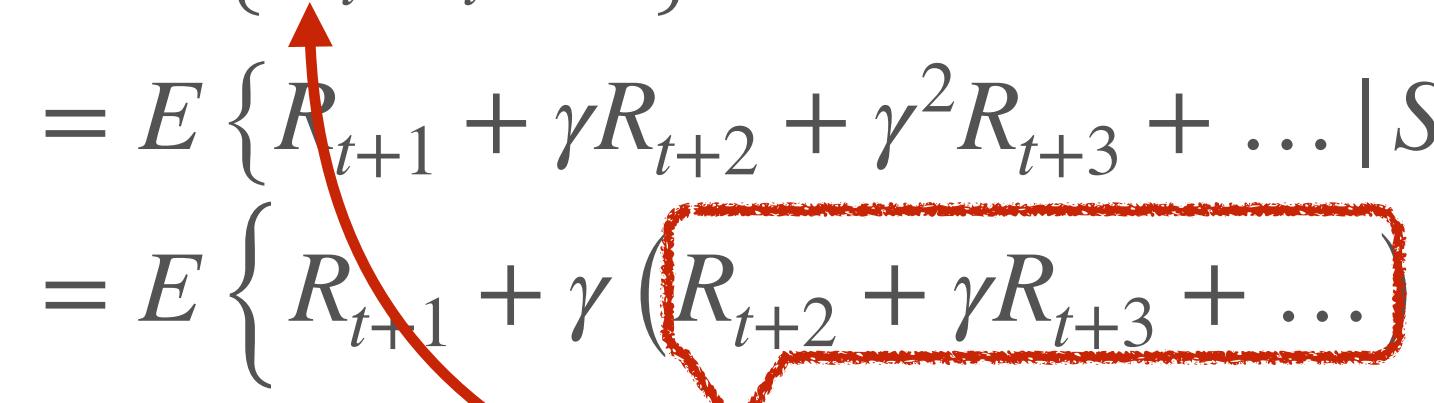
$v(s)$ for $\gamma = 1$

$$G_1 = R_2 + \gamma R_3 + \dots + \gamma^{T-2} R_T$$

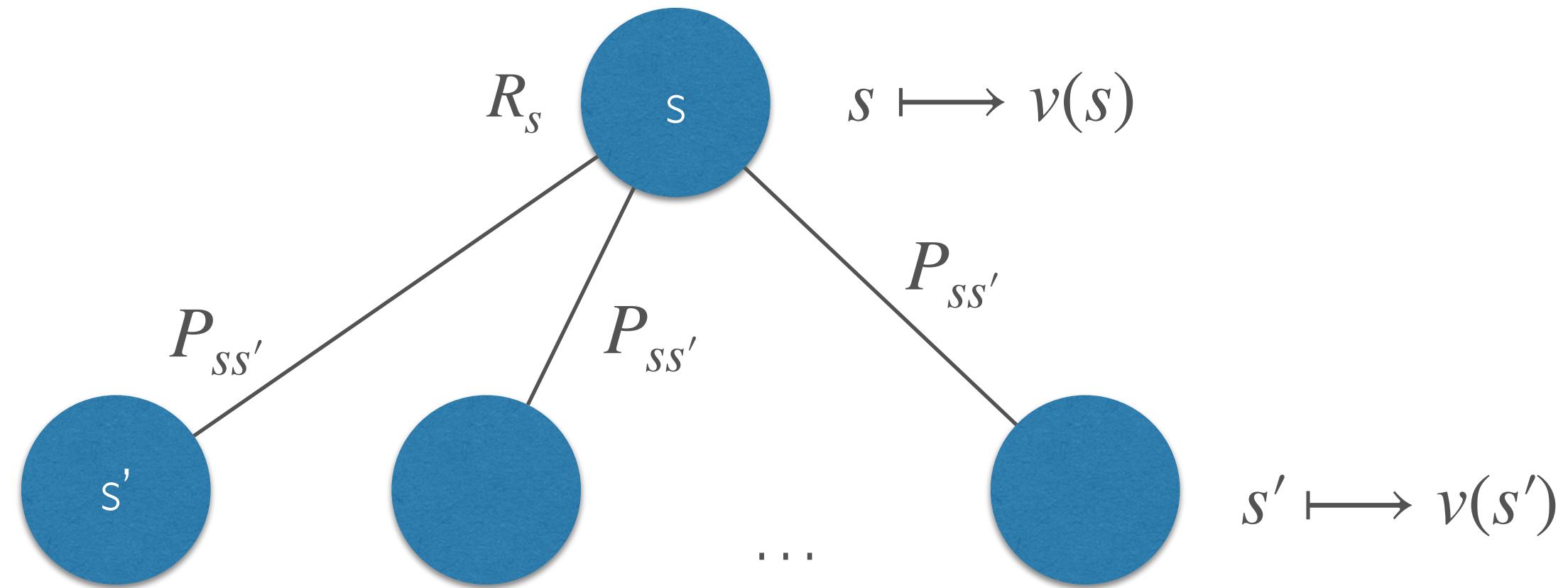


BELLMAN EQUATION FOR MRP

- Amazingly, the value function $v(s)$ can be compute **recursively** using
 - the immediate reward R_{t+1}
 - the discounted future rewards $\gamma v(S_{t+1})$

$$\begin{aligned}v(s) &= E \left\{ G_t | S_t = s \right\} \\&= E \left\{ R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots | S_t = s \right\} \\&= E \left\{ R_{t+1} + \gamma \left(R_{t+2} + \gamma R_{t+3} + \dots \right) | S_t = s \right\} \\&= E \left\{ R_{t+1} + \gamma (G_{t+1}) | S_t = s \right\} \\&= E \left\{ R_{t+1} + \gamma v(S_{t+1}) | S_t = s \right\}\end{aligned}$$


$$v(s) = E \left\{ R_{t+1} + \gamma v(S_{t+1}) \mid S_t = s \right\}$$

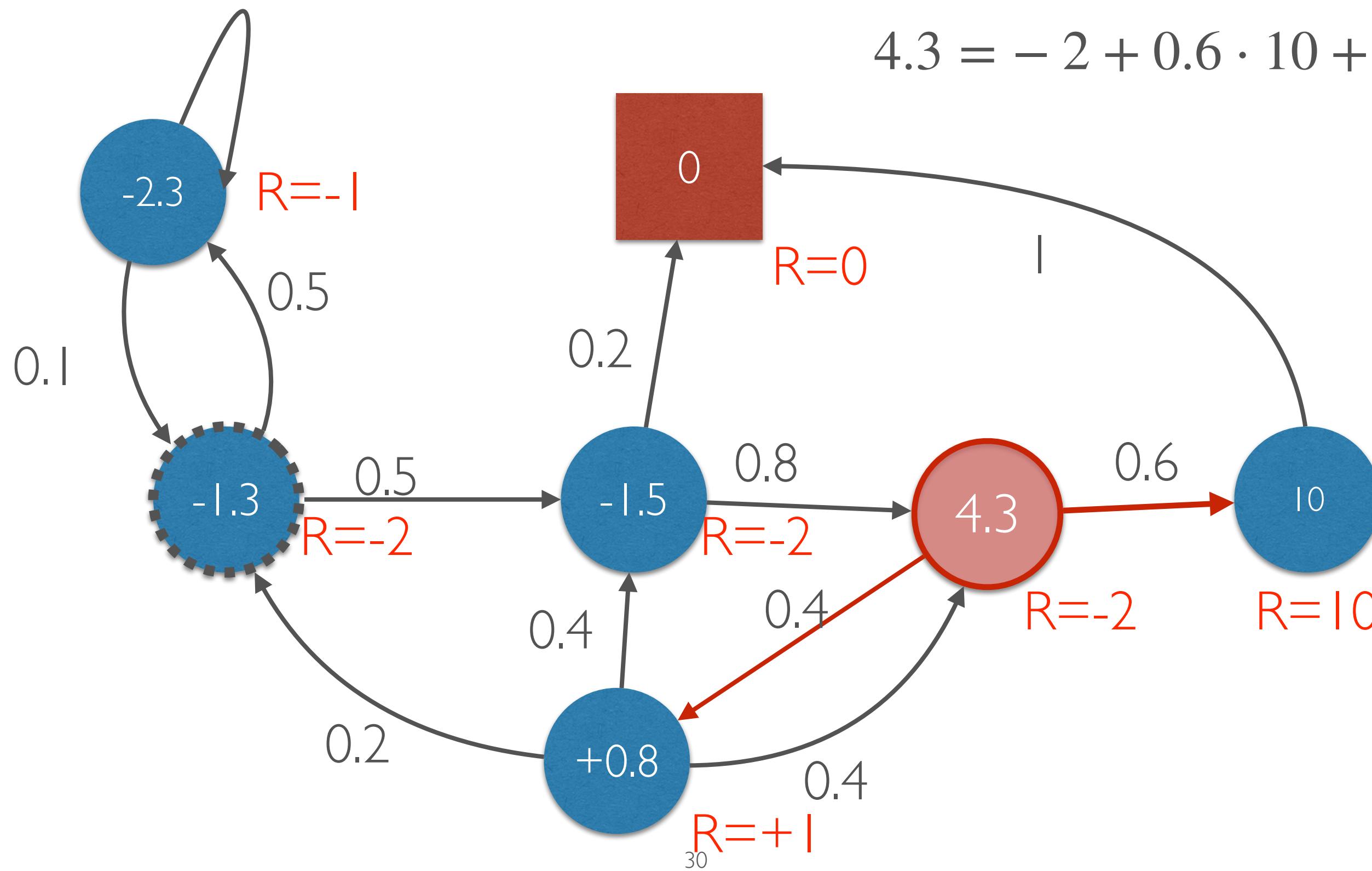


Marginalizing over the next possible states:

$$v(s) = R_s + \gamma \sum_{s' \in S} P_{ss'} v(s')$$

$$v(s) = R_s + \gamma \sum_{s' \in S} P_{ss'} v(s') \quad \gamma = 1$$

$$4.3 = -2 + 0.6 \cdot 10 + 0.4 \cdot 0.8$$



- In Markov Reward Processes, we are bound to fate, the transition matrix rules!!
- Let's extend the MRP such that we can make our own decisions!!


 S_t

what actions such that
 $v(s_t) = \max ?$


 S_{goal}


MARKOV DECISION PROCESSES



MARKOV DECISION PROCESS

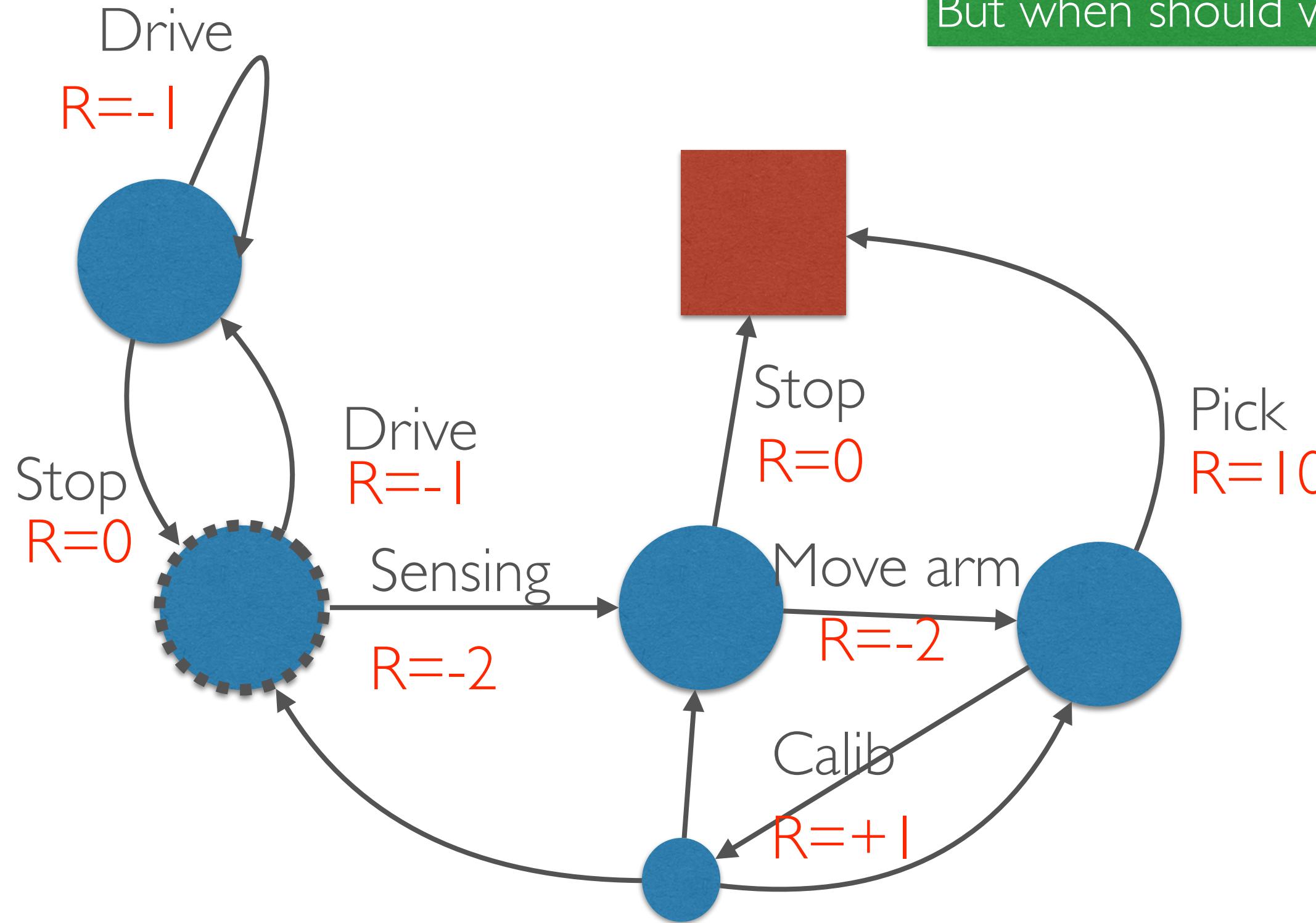
- A Markov Decision Process is a Markov Reward Process with decisions on actions

Definition: A **Markov Decision Process** is a tupel (S, A, P, R, γ)

- S is a finite set of states
- A is a finite set of actions
- P is a state transition probability matrix $P_{ss'}^a = P(S_{t+1} = s' | S_t = s, A_t = a)$
- R is a reward function $R_s^a = E(R_{t+1} | S_t = s, A_t = a)$
- γ is a discount factor $\gamma \in [0,1]$



But when should we do what action?



Policy

Value Functions



LUND
UNIVERSITY

POLICIES:WHAT TO DO WHEN

Definition: A **policy** π is a distribution over actions given states,

$$\pi(a | s) = P(A_t = a | S_t = s)$$

- Given that I am in state s what action a should I take?
- Policies define the behaviour of the agent
- Policies are *stationary*, i.e., they do not depend on the time but only on the state $S_t = s$

$$\sum_{a \in A} \pi(a | s) = 1$$



2 TYPES OF VALUE FUNCTIONS

Definition: The *state-value function* $v_\pi(s)$ of an MDP is the expected return starting from state s and following policy π

$$v_\pi(s) = E_\pi \{ G_t | S_t = s \}$$

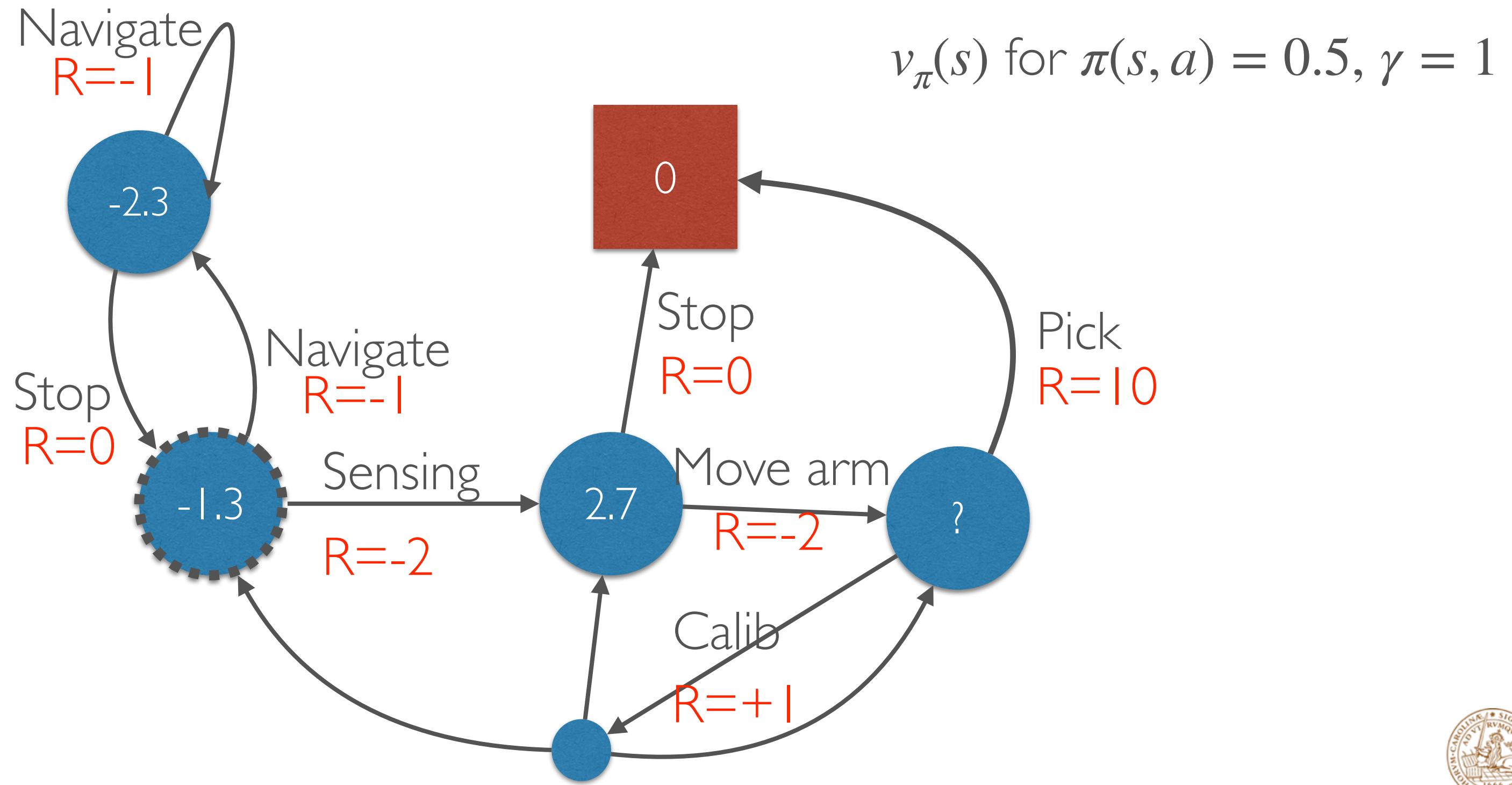
How good is it to be at LTH?

Definition: The *action-value function* $q_\pi(s, a)$ of an MDP is the expected return starting from state s by taking action a and following policy π

$$q_\pi(s, a) = E_\pi \{ G_t | S_t = s, A_t = a \}$$

Being at LTH how valuable is it to take a class a ?

STATE-VALUE FUNCTION FOR PICKING



BELLMAN EXPECTATION EQUATION

- The state-value function and the action value function can again be recursively computed

$$v_{\pi}(s) = E_{\pi} \{ R_{t+1} + \gamma v_{\pi}(S_{t+1}) \mid S_t = s \}$$

$$q_{\pi}(s, a) = E_{\pi} \{ R_{t+1} + \gamma q_{\pi}(S_{t+1}, A_{t+1}) \mid S_t = s, A_t = a \}$$

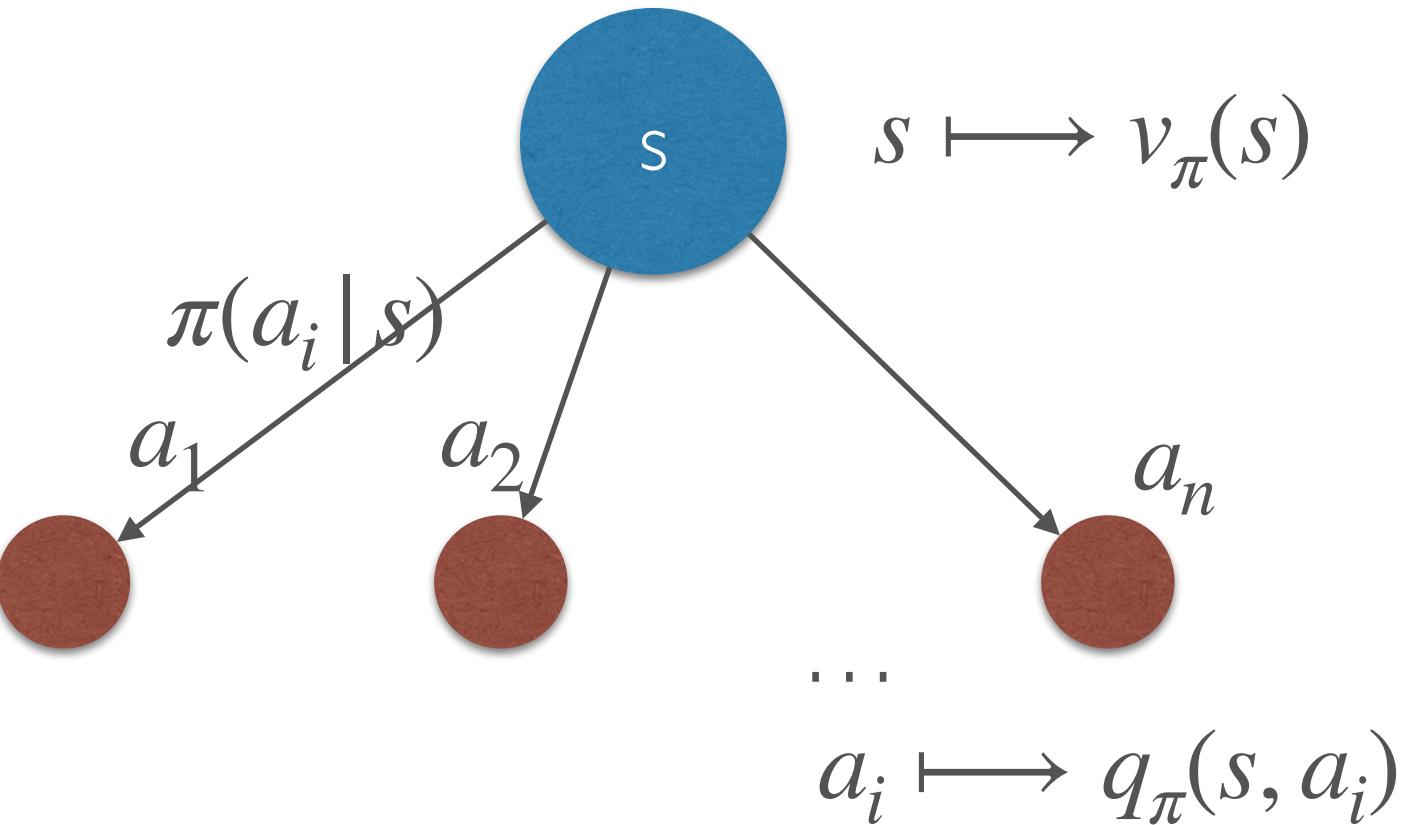
Proof works exactly the same as for $v(s)$



LUND
UNIVERSITY

BELLMAN EXPECTATION EQUATION

- From state s , we can take different actions $a_i \in A$ chosen according to $\pi(a | s)$



Averaging / marginalising over the action a

$$v_\pi(s) = \sum_{a \in A} \pi(a | s) q_\pi(s, a)$$

The choice is yours!

At LTH, choose a course a



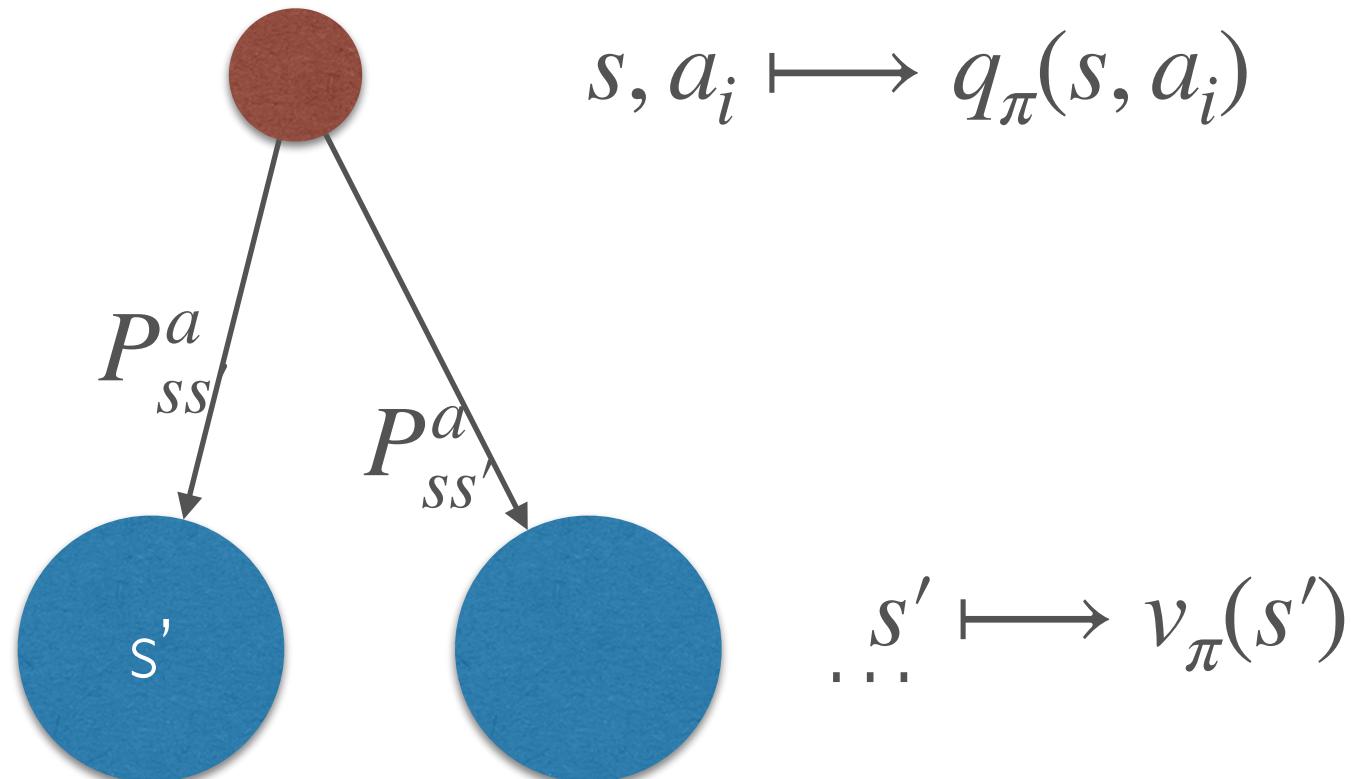
LUND
UNIVERSITY

BELLMAN EXPECTATION EQUATION

- Based on state s and action a we get a reward R_s^a

$$q_{\pi}(s, a) = R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a v_{\pi}(s')$$

from here, it is fate....



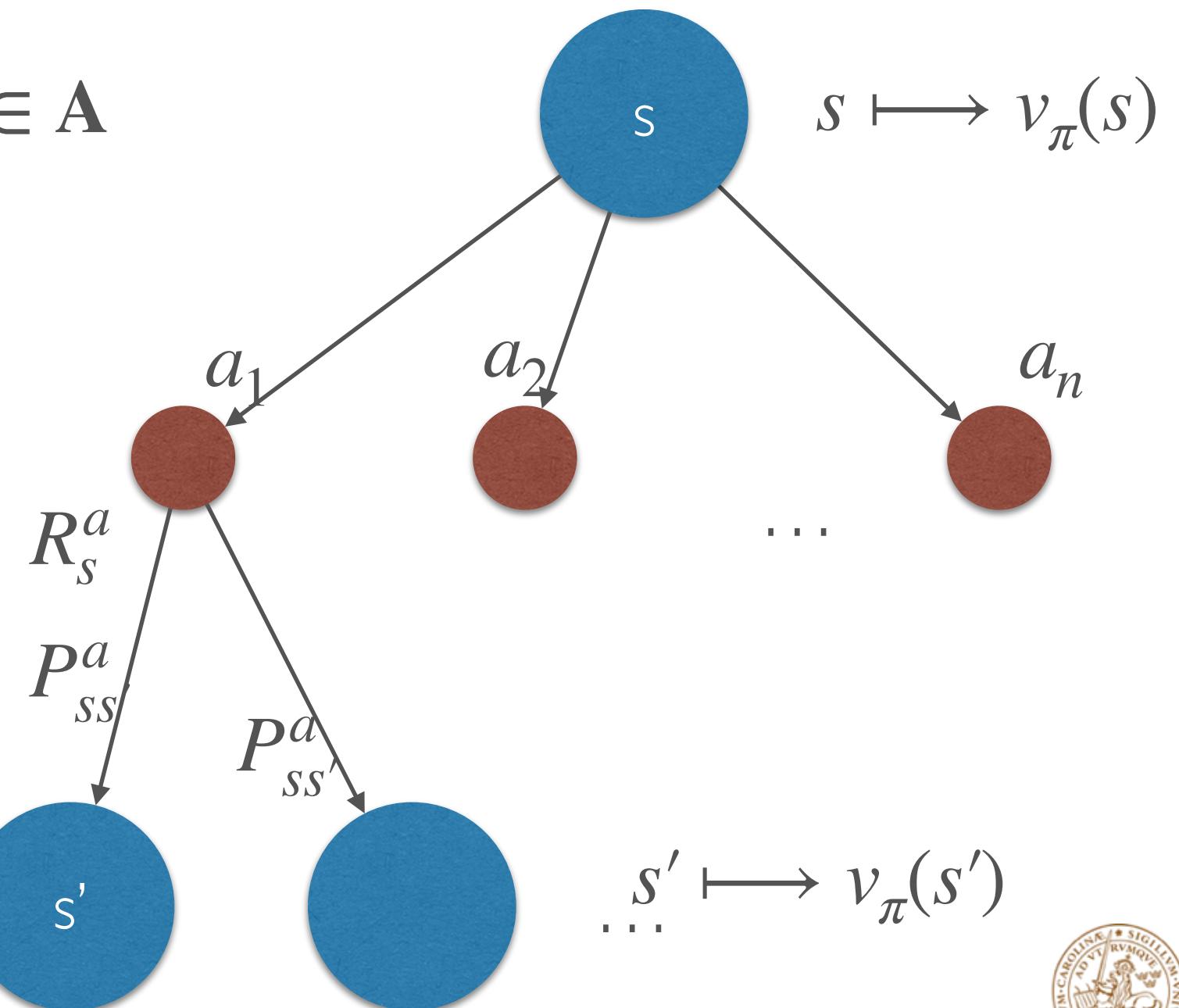
teacher and students can be good / bad; pandemic can strike, ...



BELLMAN EXPECTATION EQUATION

- From state s , we can take different actions $a_i \in A$ chosen according to $\pi(a | s)$
 - Based on s and a we get a reward R_s^a

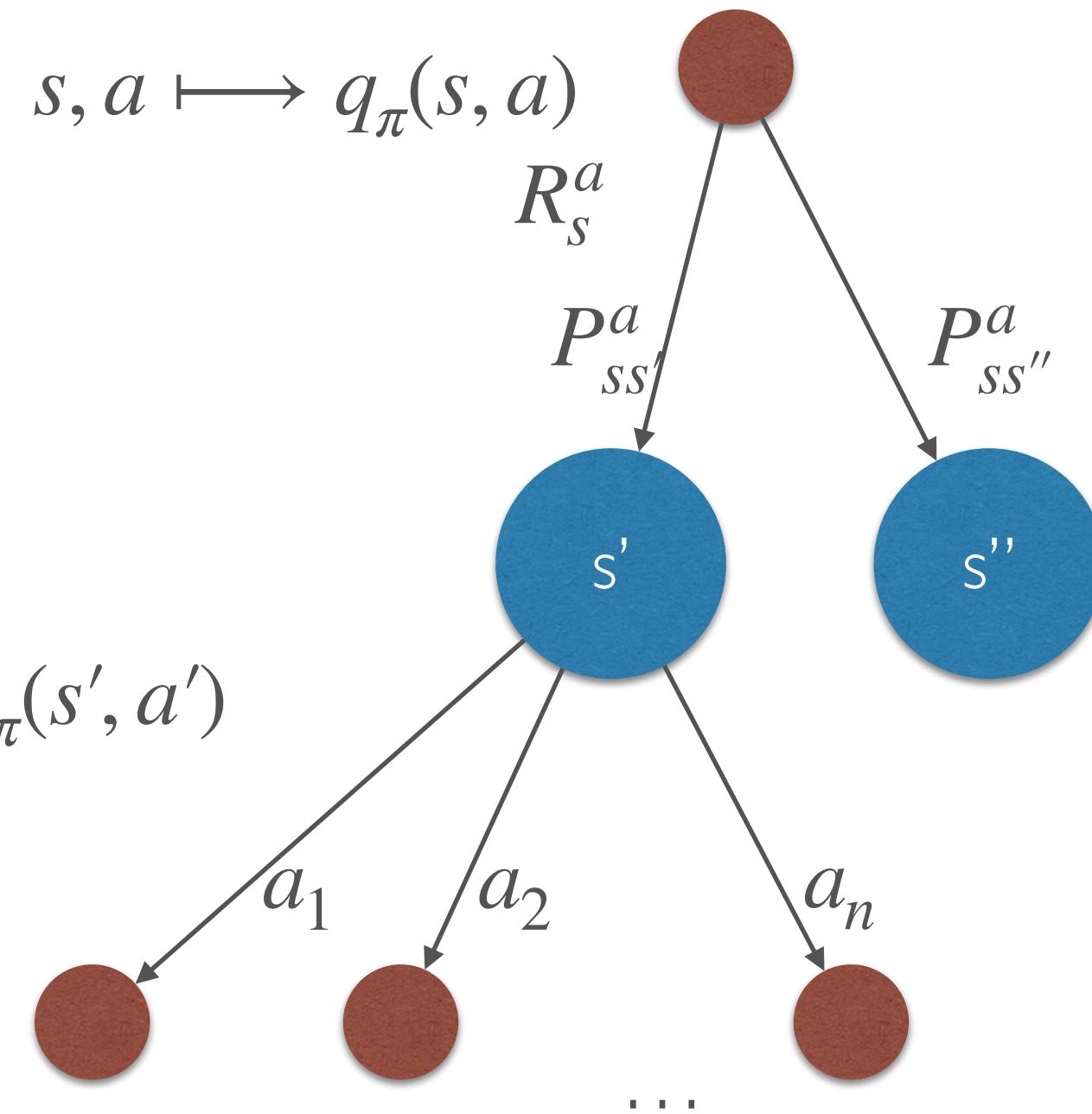
$$v_\pi(s) = \sum_{a \in A} \pi(a | s) \left(R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a v_\pi(s') \right)$$



BELLMAN EXPECTATION EQUATION

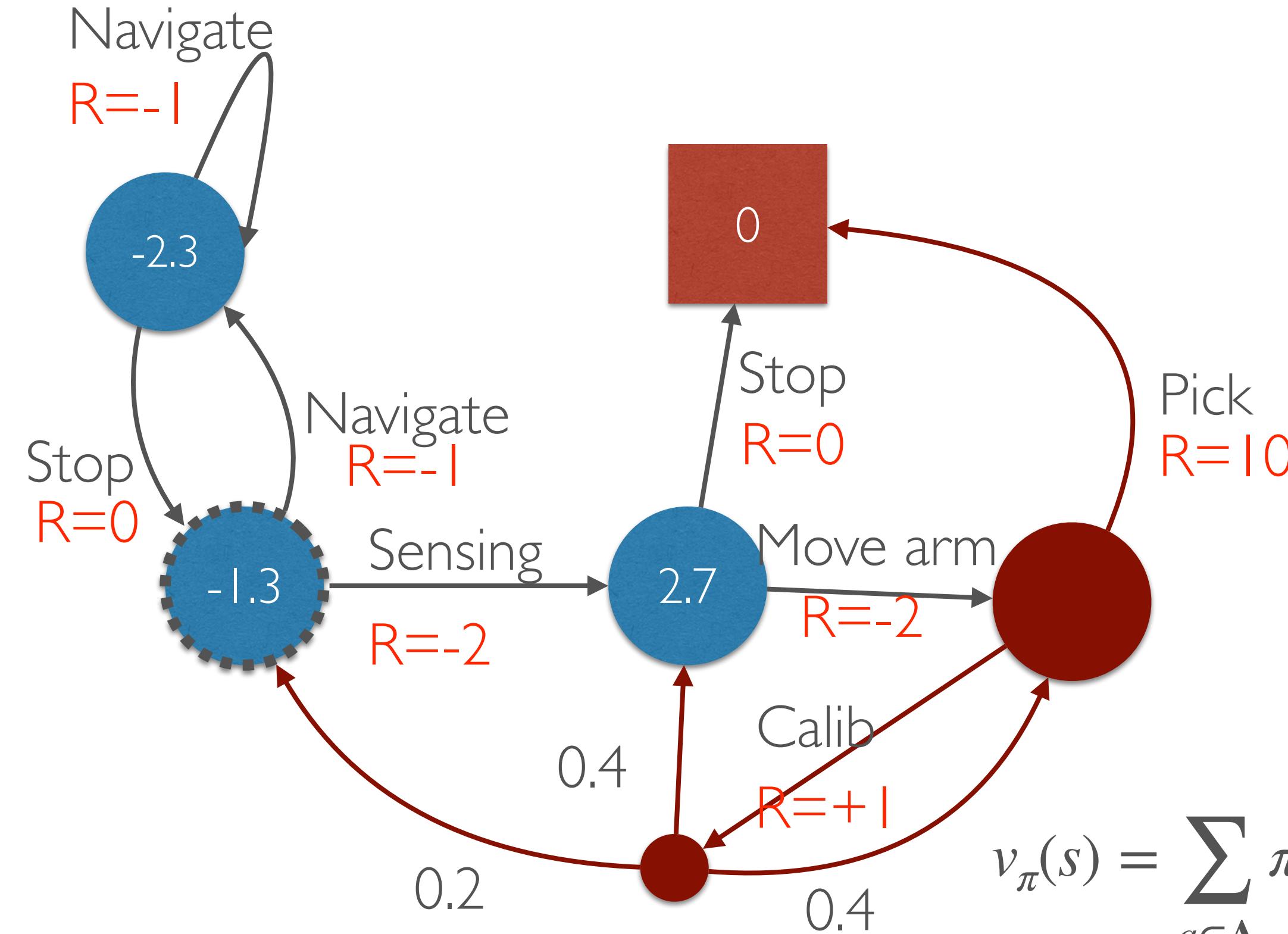
$$q_{\pi}(s, a) = R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a \sum_{a' \in A} \pi(a' | s') q_{\pi}(s', a')$$

$$s', a_i \longmapsto q_{\pi}(s', a_i)$$



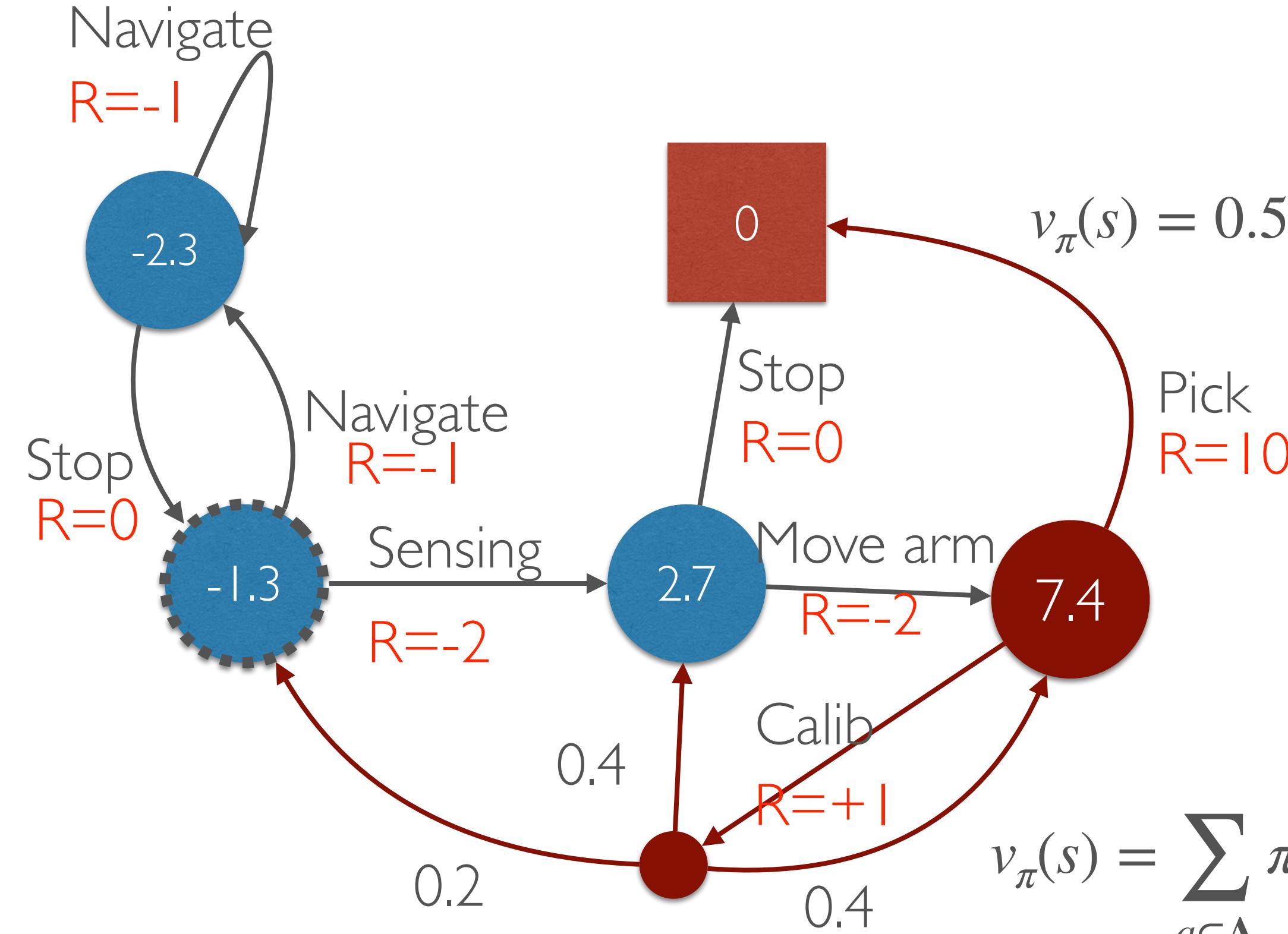
BELLMAN EXPECTATION EQUATION

$v_\pi(s)$ for $\pi(s, a) = 0.5, \gamma = 1$



$$v_\pi(s) = \sum_{a \in A} \pi(a | s) \left(R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a v_\pi(s') \right)$$

BELLMAN EXPECTATION EQUATION



$$v_\pi(s) \text{ for } \pi(s, a) = 0.5, \gamma = 1$$

$$v_\pi(s) = \sum_{a \in A} \pi(a | s) \left(R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a v_\pi(s') \right)$$

OPTIMAL VALUE FUNCTION

- So far, we have asked: Given a policy π what are the state values.
- We can also aim to optimise the policy π to maximise the state value function

$$v^*(s) = \max_{\pi} v_{\pi}(s)$$

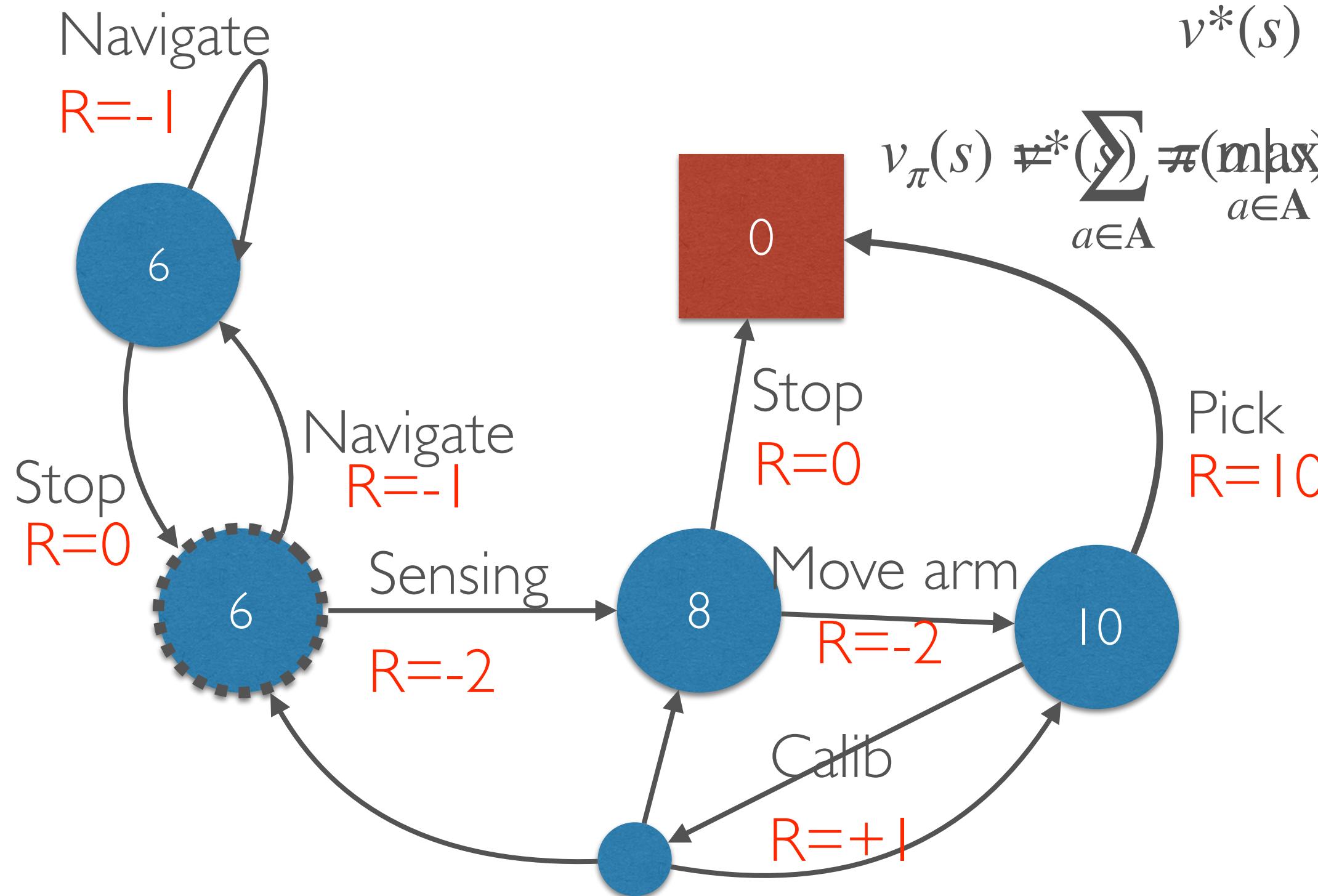
Definition: $v^*(s)$ is called the ***optimal state-value*** function

- Likewise, we can find the π that optimises the action-value function

$$q^*(s, a) = \max_{\pi} q_{\pi}(s, a)$$

Definition: $q^*(s, a)$ is called the ***optimal action-value function***

EXAMPLE: OPTIMAL STATE-VALUE FUNCTION FOR PICKING



$v^*(s)$ for $\gamma = 1$

$$v_\pi(s) \leftarrow \max_{a \in A} \left(R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a v_\pi(s') \right)$$

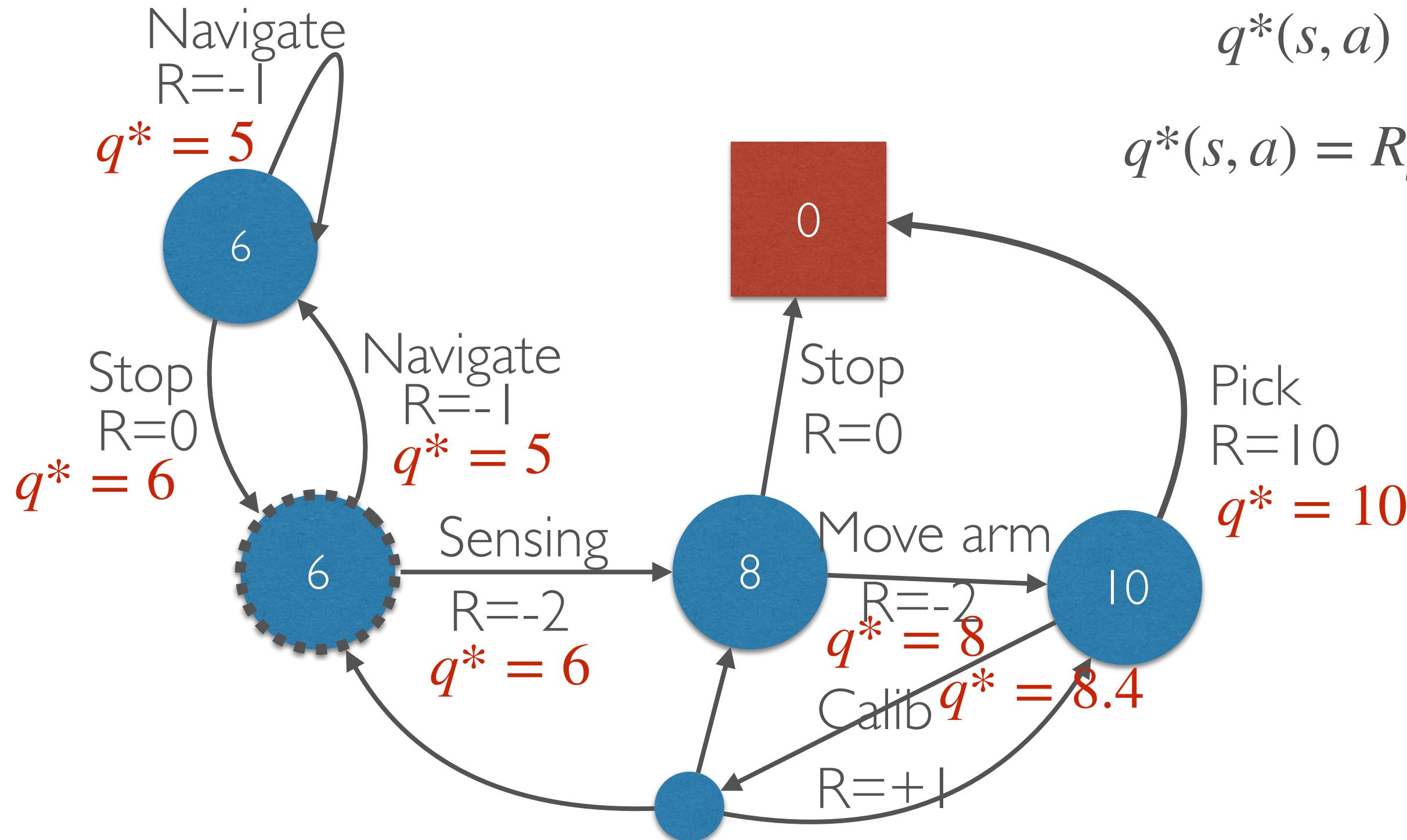
Pick
 $R=10$

Move arm
 $R=-2$

Calib
 $R=+1$



OPTIMAL ACTION-VALUE FUNCTION FOR PICKING



OPTIMAL POLICY

Define a partial ordering over policies:

$$\pi \geq \pi' \text{ if } v_\pi(s) \geq v_{\pi'}(s) \forall s$$

- **Theorem:** For every Markov Decision Process
 - I. There exists an optimal policy π^* that is better than or equal to all other policies
$$\pi^* \geq \pi \quad \forall \pi$$
 2. All optimal policies achieve the optimal value function $v_{\pi^*}(s) = v^*(s)$
 3. All optimal policies achieve the optimal action-value function $q_{\pi^*}(s, a) = q^*(s, a)$



FINDING AN OPTIMAL POLICY

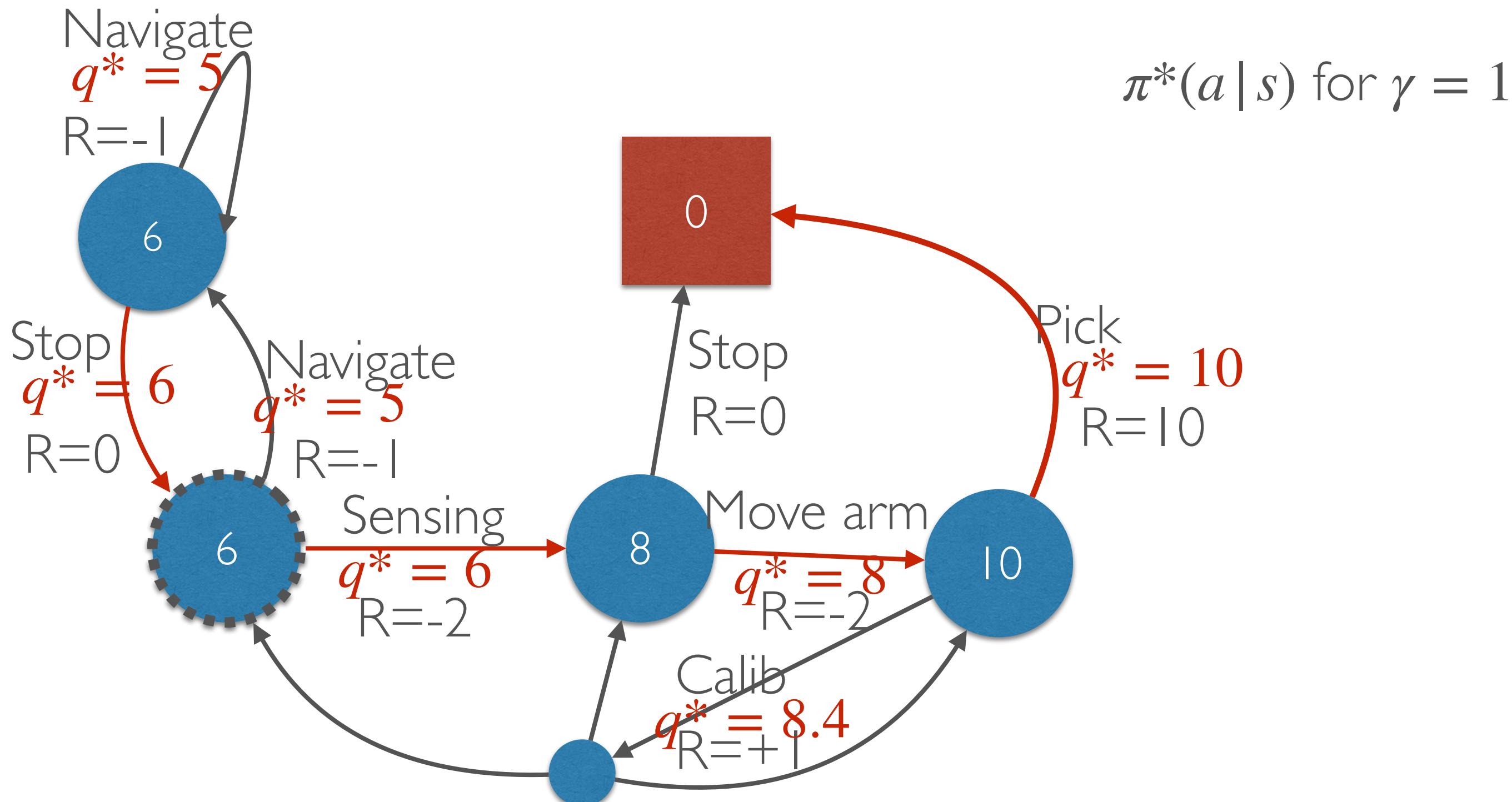
- An optimal policy can be found by maximising over $q^*(s, a)$

$$\pi^*(a | s) = \begin{cases} 1 & \text{if } a = \arg \max_{a \in A} q^*(s, a) \\ 0 & \text{otherwise} \end{cases}$$

- There is always a deterministic optimal policy for any MDP
- If we know $q^*(s, a)$, we immediately have the optimal policy.

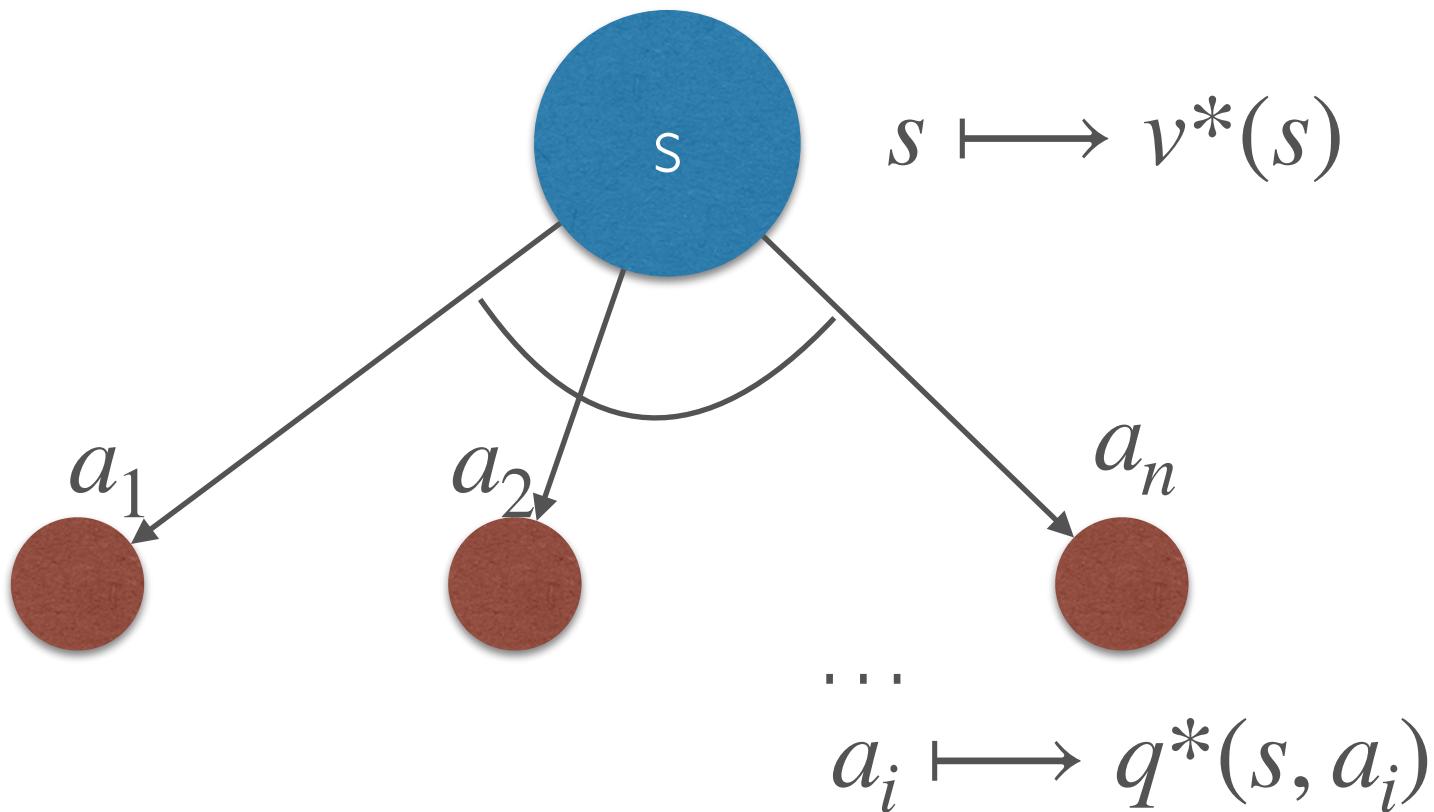


LUND
UNIVERSITY



BELLMAN OPTIMALITY EQUATION

- From state s , we take action $a_i \in \mathbf{A}$ that has maximal value $q(s, a)$



Maximizing over a $v^*(s) = \max_a q^*(s, a)$

The choice is ours!

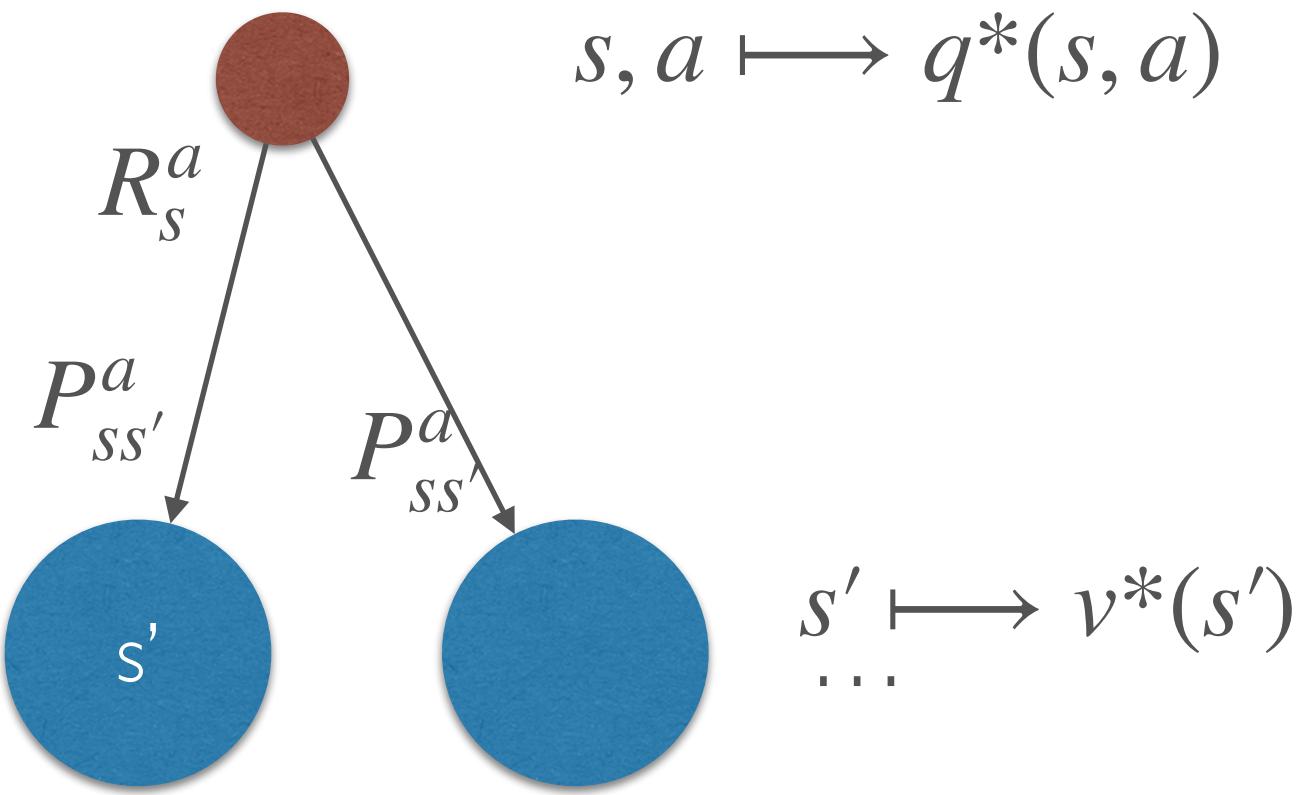


LUND
UNIVERSITY

BELLMAN OPTIMALITY EQUATION

- Based on state s and action a we get a reward R_s^a

$$q^*(s, a) = R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a v^*(s')$$

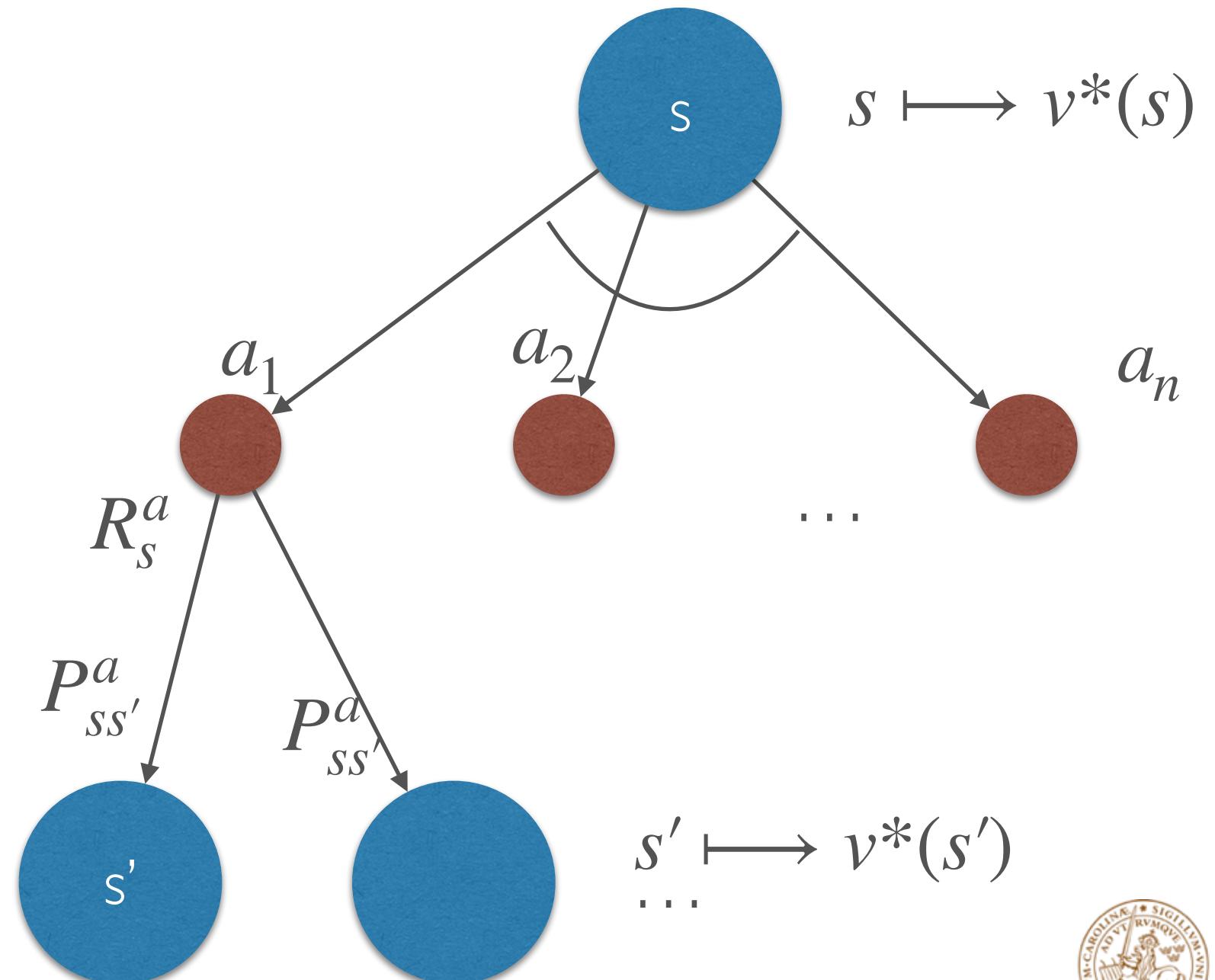


from here, it is fate again...

BELLMAN OPTIMALITY EQUATION

- From state s , we choose action $a_i \in A$ that gives maximal reward $q(s, a)$
- Based on s and a we get a reward R_s^a

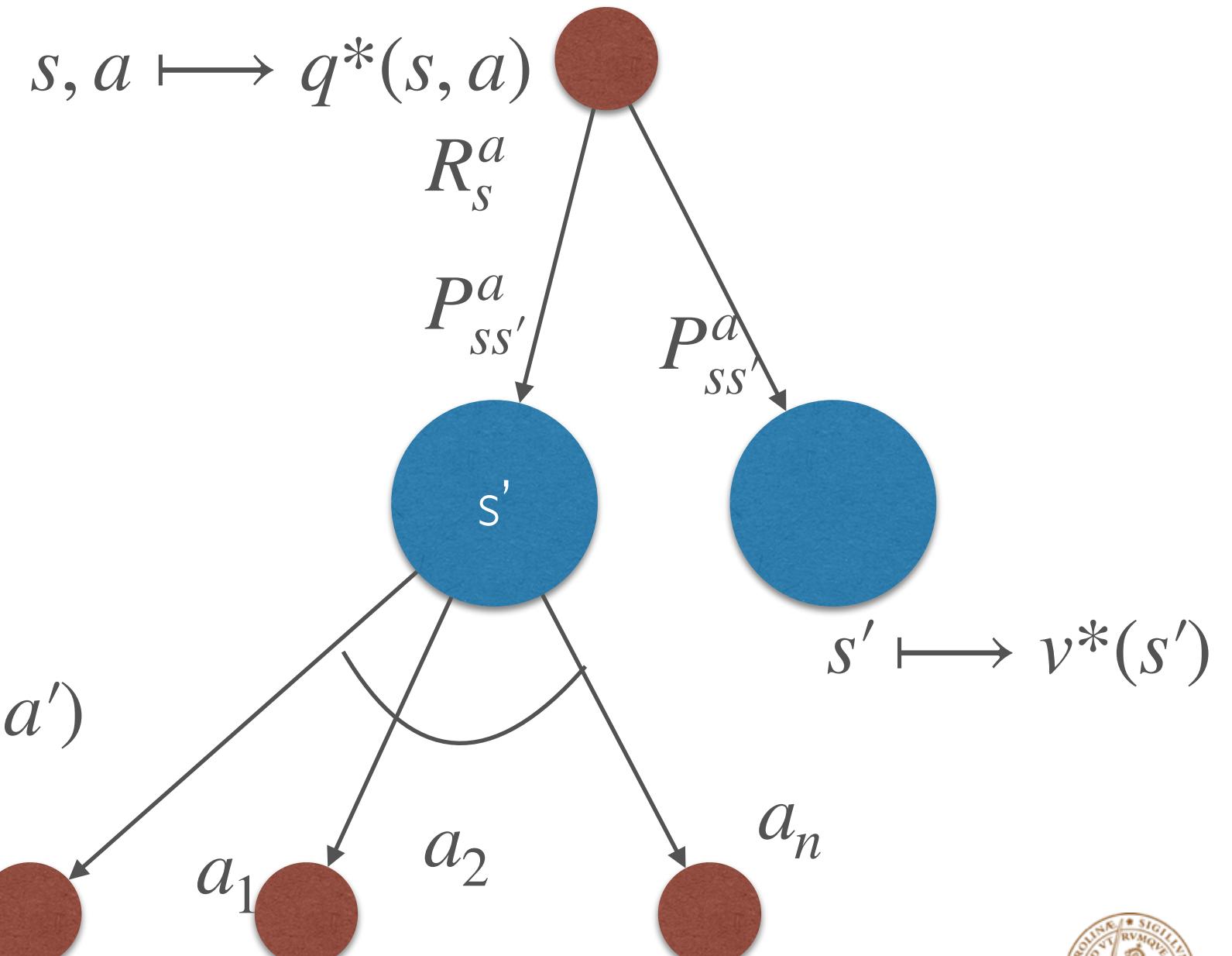
$$v_*(s) = \max_{a \in A} \left(R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a v^*(s') \right)$$



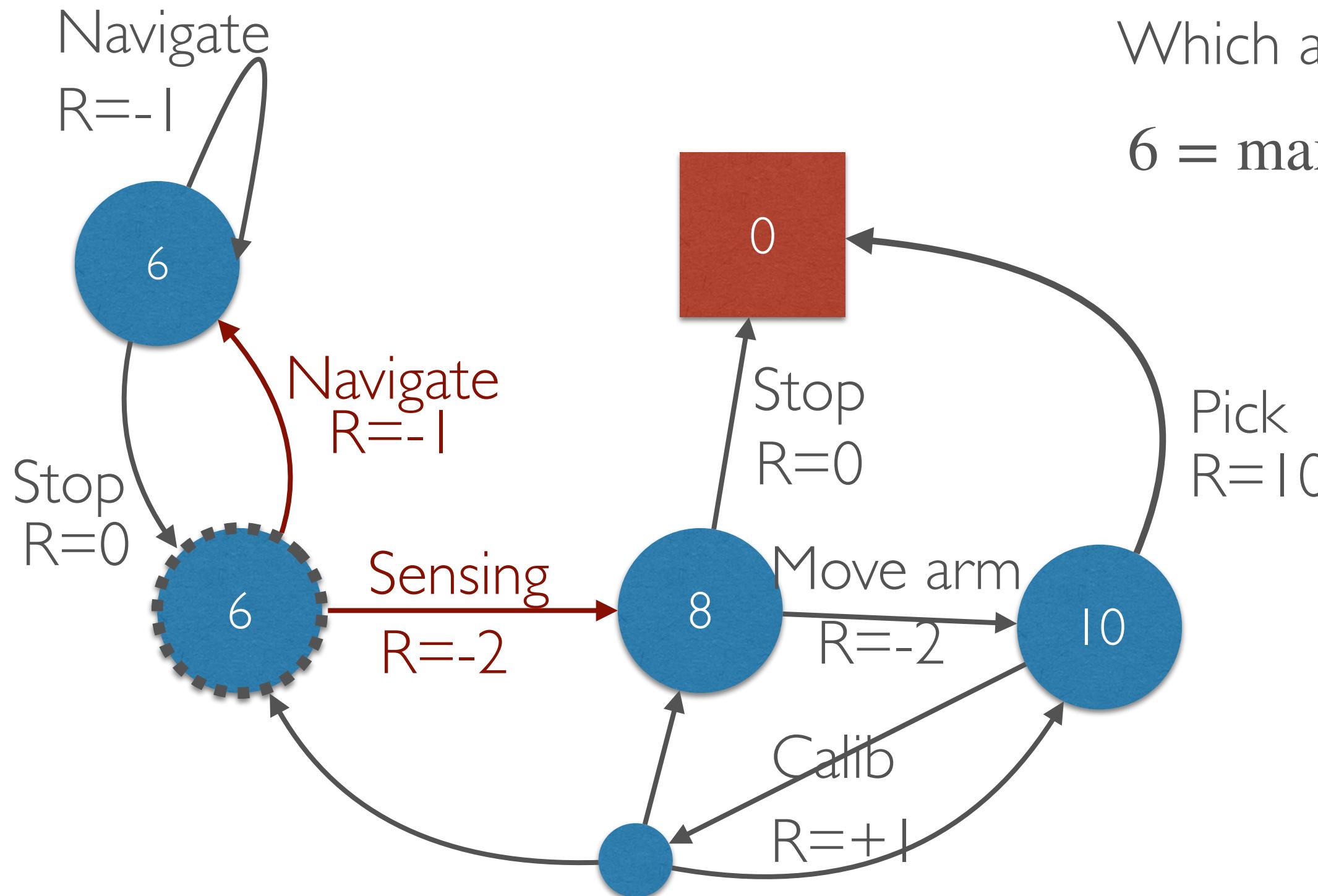
BELLMAN OPTIMALITY EQUATION

$$q^*(s, a) = R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a \max_{a' \in A} q^*(s', a')$$

$$s', a' \mapsto q^*(s', a')$$



BELLMAN OPTIMALITY EQUATION



Which action should we take?

$$6 = \max\{-2 + 8, -1 + 6\}$$



SOLVING THE BELLMAN OPTIMALITY EQUATION

- Bellman Optimality Equation is non-linear
- No closed form solution (in general)
- Many iterative solution methods
 - Dynamic Programming: Value Iteration, Policy Iteration
 - Q-learning
 - Sarsa

This is setting the stage for Reinforcement Learning



SUMMARY

- Markov Model, Markov Chain
- Markov Reward Process, Bellman Equation for the state value function
- Markov Decision Process, Bellman Eq. for state value and action value functions
- Identify the optimal policy, state value function and action value function



NEXT TIME

- how to find the optimal
 - state value functions,
 - action value functions
 - optimal policies
 - with and without a model (model-free)



LUND
UNIVERSITY

QUESTIONS?

