Markov decision processes (MDPs)

Kevin Lloyd

Department of Computational Neuroscience
MPI for Biological Cybernetics

klloyd@tue.mpg.de

decision-making problems

- one-shot/single-stage/'static': current decision just affects immediate outcome

- sequential/multi-stage/'dynamic': current decision affects immediate outcome but also *context in which future decisions are made* (and thus future outcomes)

A **Markov decision process (MDP)** is a formal model of sequential decision-making

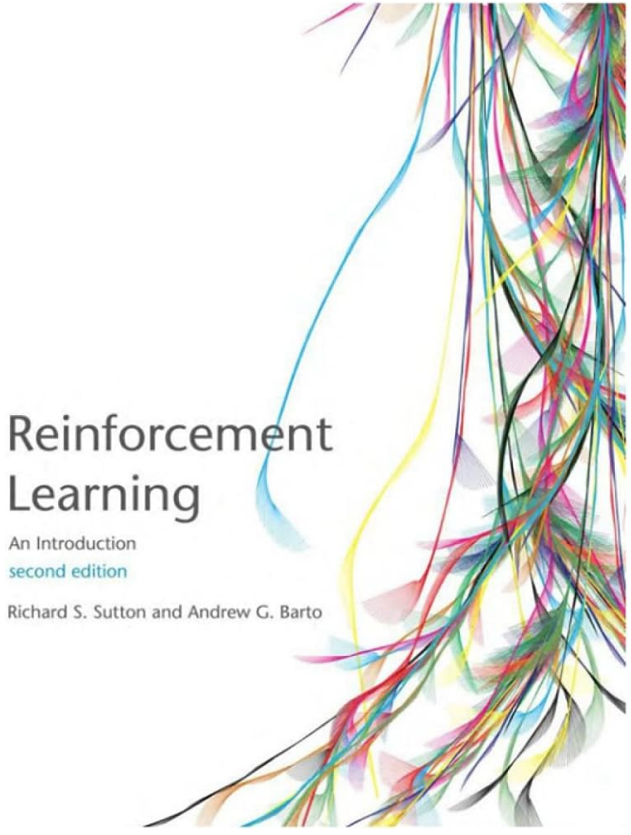
- **flexible**: many decision problems can be formulated as MDPs

- **solvable** (in principle): dynamic programming

- **foundational**: e.g., for understanding reinforcement learning methods

decision-making problems

. . . all of these [RL] methods can be viewed as attempts to achieve much the same affect as DP [Dynamic Programming], only with less computation and without assuming a perfect model of the environment. (Sutton & Barto, 2018, p.73)



[**Chapters 3 & 4]

decision-making problems

- one-shot/single-stage/'static': current decision just affects immediate outcome

- sequential/multi-stage/'dynamic': current decision affects immediate outcome but also *context in which future decisions are made* (and thus future outcomes)

A **Markov decision process (MDP)** is a formal model of sequential decision-making

→ evidence of altered decision-making in psychiatric populations

- **flexible**: many decision problems can be formulated as MDPs

- **solvable** (in principle): dynamic programming

- **foundational**: e.g., for understanding reinforcement learning methods

→ formal framework for thinking about what might lead to these alterations [cf. Huys et al. (2015), 'Decision-theoretic psychiatry']

outline

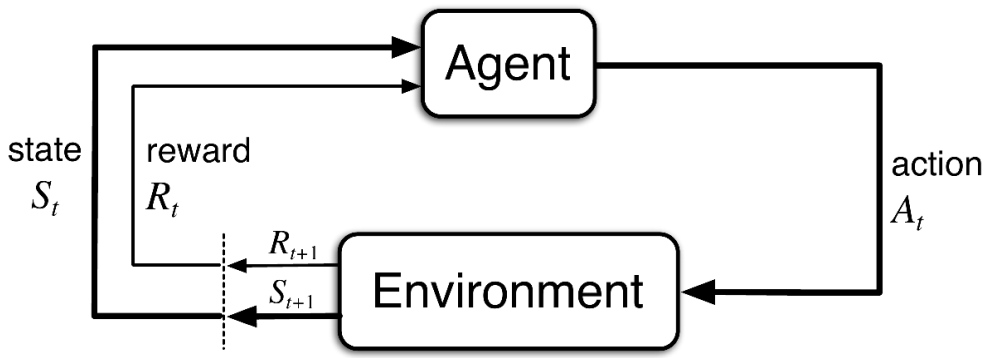
1. MDP definition
2. Solution methods (dynamic programming)
3. Comments/Extensions

MDP

- set of states $\mathcal{S}$
- set of actions $\mathcal{A}$
- set of rewards $\mathcal{R}$
- dynamics function p:

$$p(s', r|s, a) := \Pr(\underbrace{S_{t+1} = s', R_{t+1} = r}_{\text{next state \& reward}} \mid \underbrace{S_t = s, A_t = a}_{\text{current state \& action}}),$$

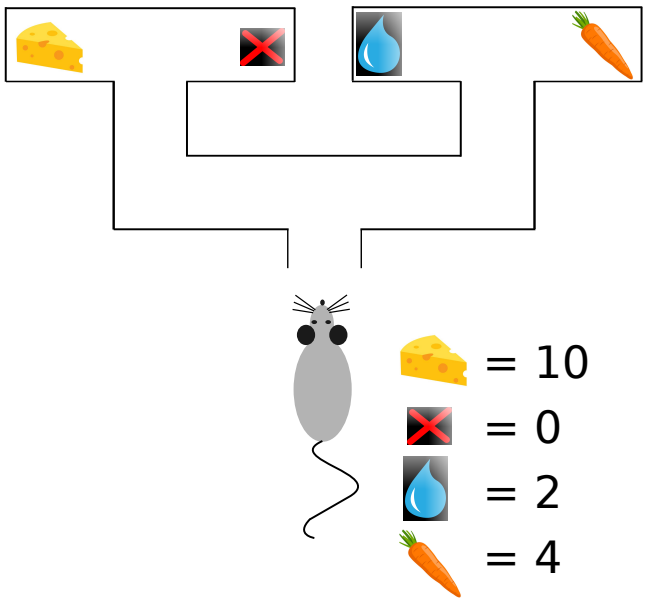
for all $s, s' \in \mathcal{S}$, $r \in \mathcal{R}$, $a \in \mathcal{A}$

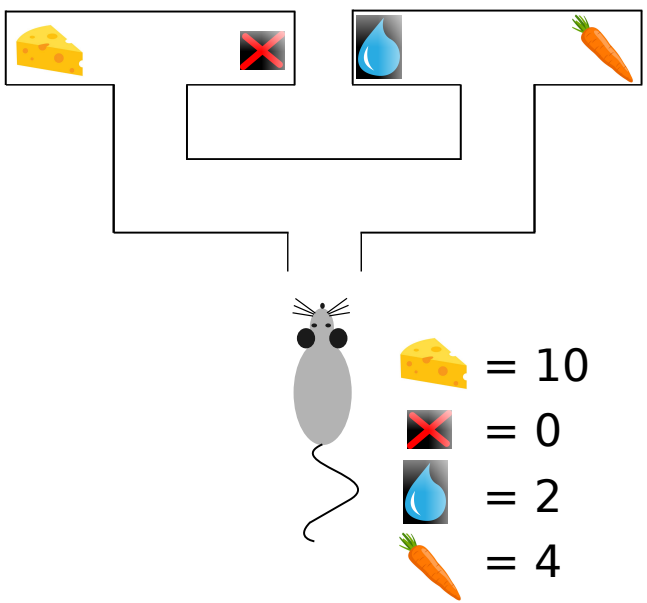


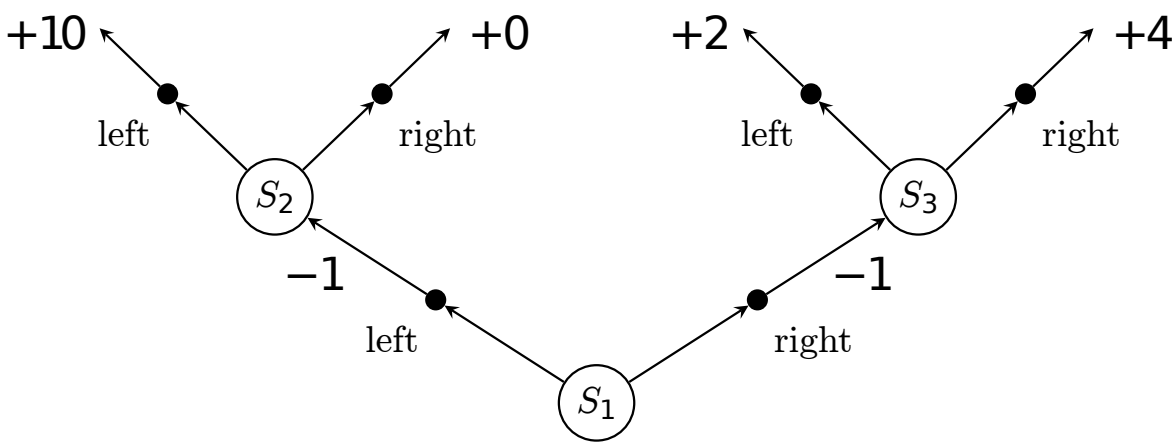
(Sutton & Barto, 2018)

Markov property: 'future is independent of the past given the present'
(i.e., probability of next state and reward depend only on current state and action)

MDP


= 10
= 0
= 2
= 4


= 10
= 0
= 2
= 4


+10
+0
+2
+4
left
right
left
right
S_2
S_3
−1
−1
left
right
S_1

MDP

Objective: find a 'policy' that maximizes a long-run measure of reward

- **policy,** π
  - for each state, the probability of taking each possible action

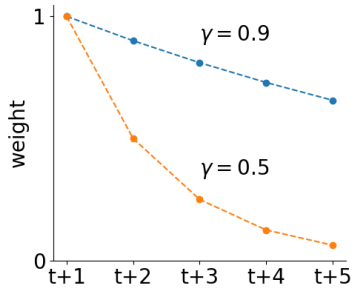
$$\pi(a|s) \doteq \Pr(A_t = a|S_t = s)$$

- **return,** G_t
  - total reward from time-step t (episodic)

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

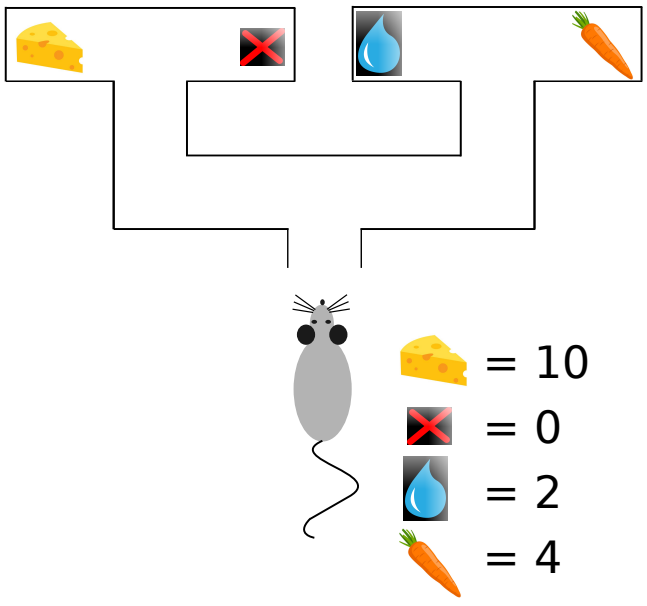
  - total discounted reward from time-step t (continuing)

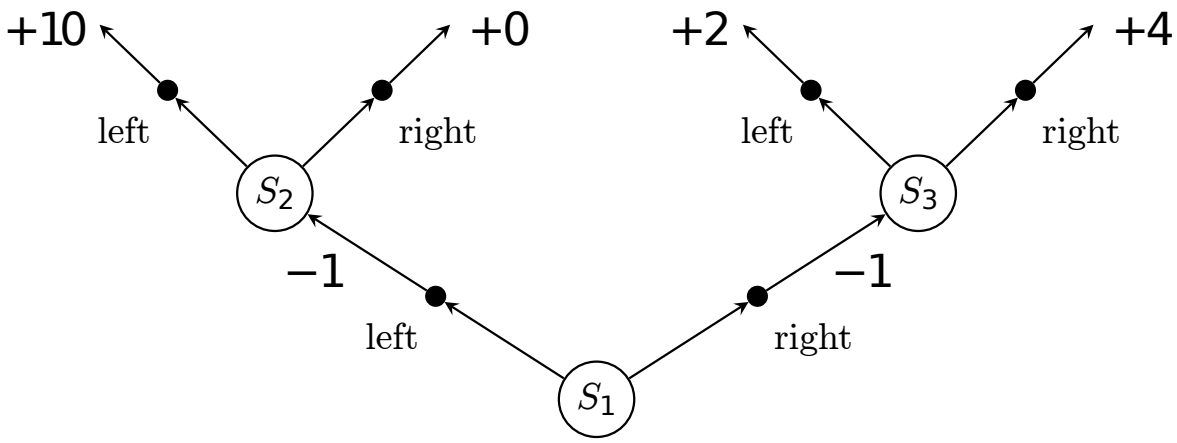
$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$



Obj.: find a policy that maximizes the **expected** (i.e., average) return

simple example





Objective: find a policy π that maximizes the expected undiscounted ($\gamma = 1$) return for a single trial

interim summary

So far:

- defined an MDP: $\langle \mathcal{S}, \mathcal{A}, \mathcal{R}, p \rangle$
- specified an objective: maximize expected return

. . . so how to actually solve it?!

value functions

Key move: define **value function(s)**

- **state value function**: how good is it to be in a state, given that I am following a policy π?

$$v_\pi(s) \doteq \mathbb{E}_\pi[G_t|S_t = s]$$

- **state-action value function**: how good is to be in a state *and perform a particular action*, given that I am (otherwise) following policy π?

$$q_\pi(s, a) \doteq \mathbb{E}_\pi[G_t|S_t = s, A_t = a]$$

value functions

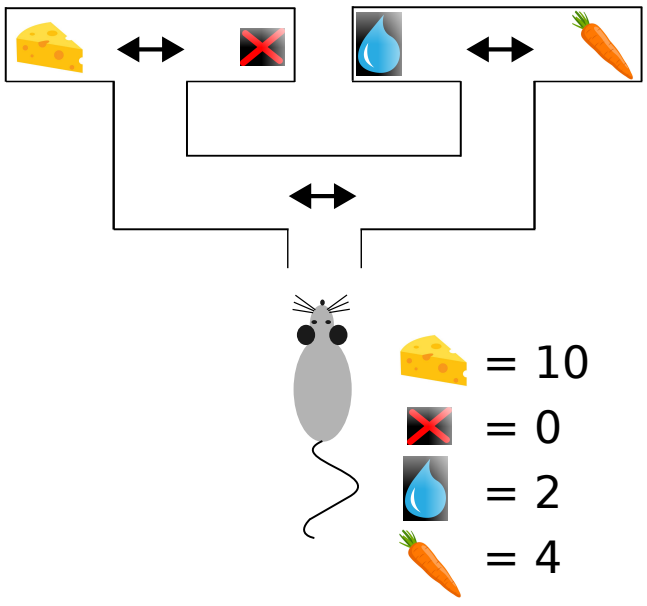
Key observation: a state's value is related to that of its successor states (local 'consistency' condition)

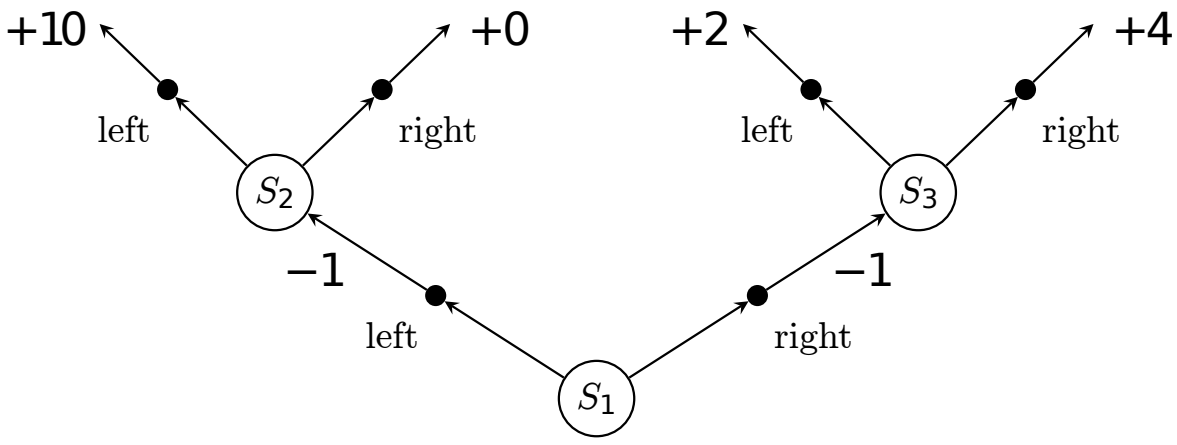
$$
\begin{aligned}
v_\pi(s) &\doteq \mathbb{E}_\pi[G_t|S_t = s] \\
&= \mathbb{E}_\pi[\underbrace{R_{t+1}}_{\text{immediate reward}} + \underbrace{\gamma v_\pi(s')}_{\text{discounted value of successor state}} |S_t = s] \\
&= \sum_a \pi(a|s) q_\pi(s,a)
\end{aligned}
$$

$$
\begin{aligned}
q_\pi(s,a) &\doteq \mathbb{E}_\pi[G_t|S_t = s, A_t = a] \\
&= \mathbb{E}_\pi[R_{t+1} + \gamma v_\pi(s')|S_t = s, A_t = a]
\end{aligned}
$$

'Bellman (expectation) equations'

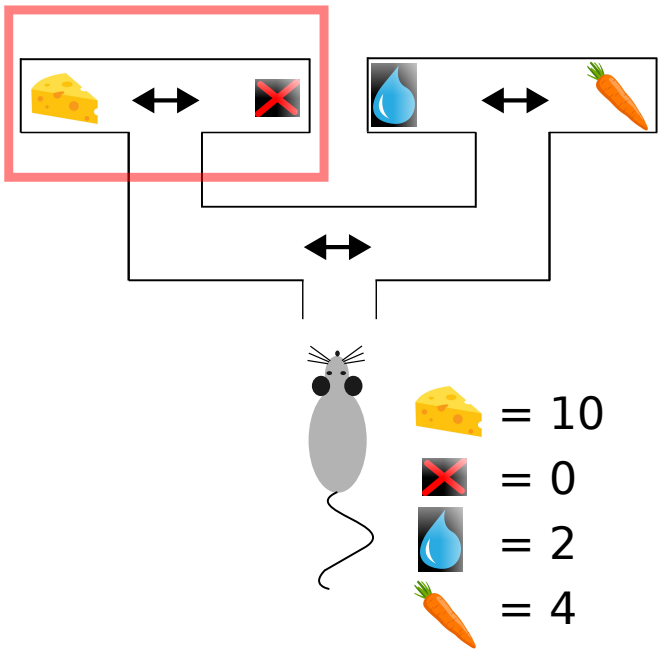
simple example: random policy

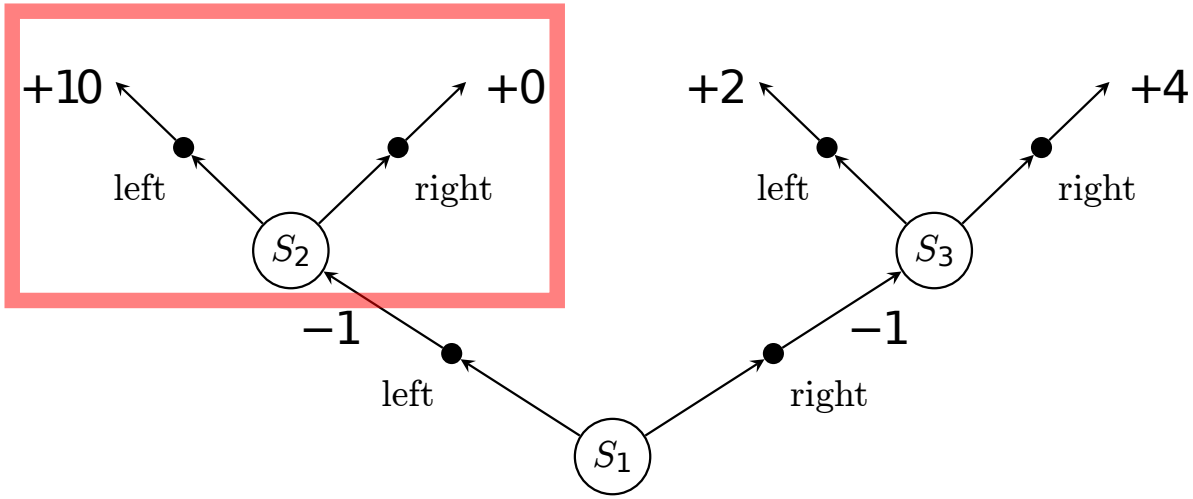


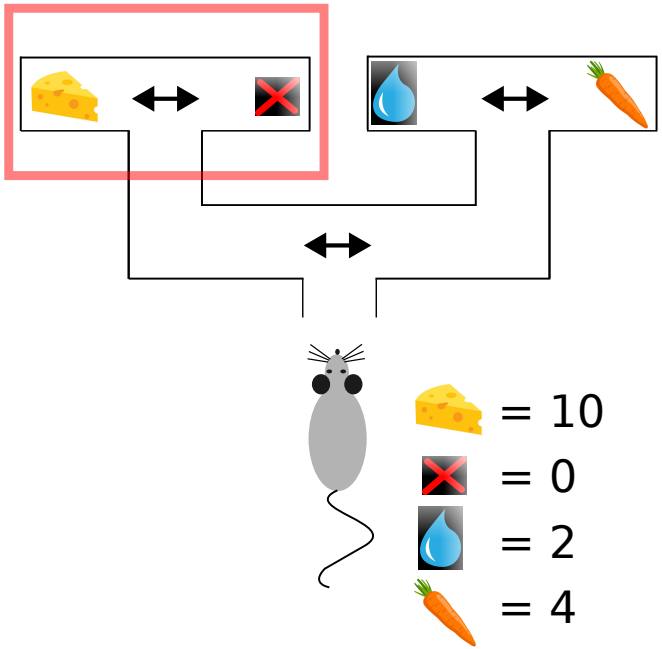


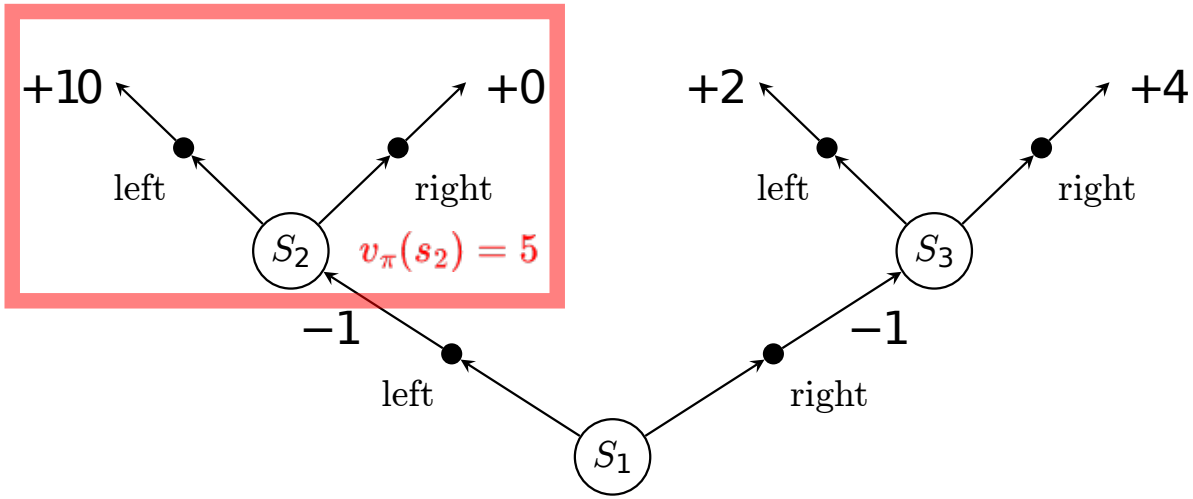
$$\pi(a = \text{left}|s_1) = \pi(a = \text{right}|s_1) = 1/2$$
$$\pi(a = \text{left}|s_2) = \pi(a = \text{right}|s_2) = 1/2$$
$$\pi(a = \text{left}|s_3) = \pi(a = \text{right}|s_3) = 1/2$$

simple example: random policy

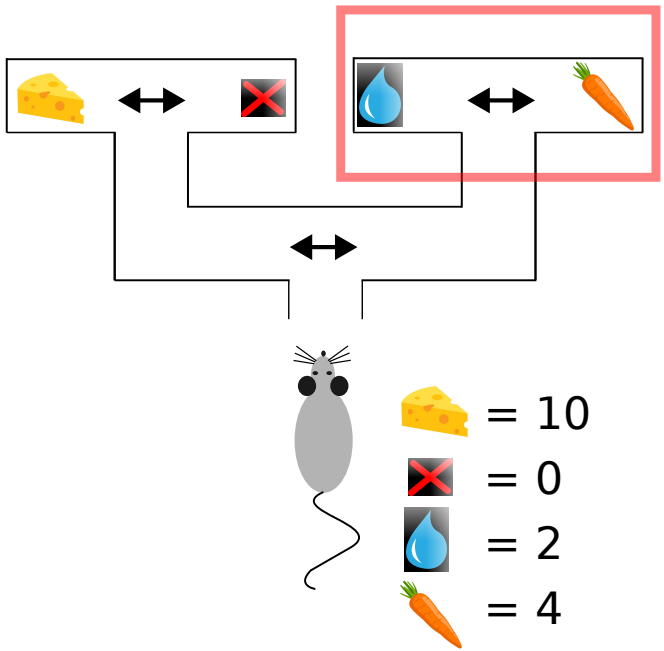


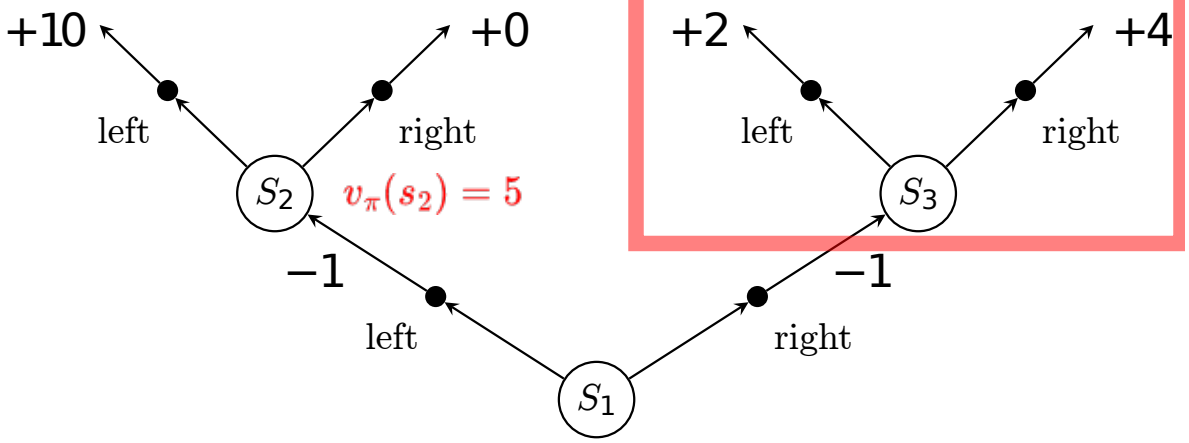


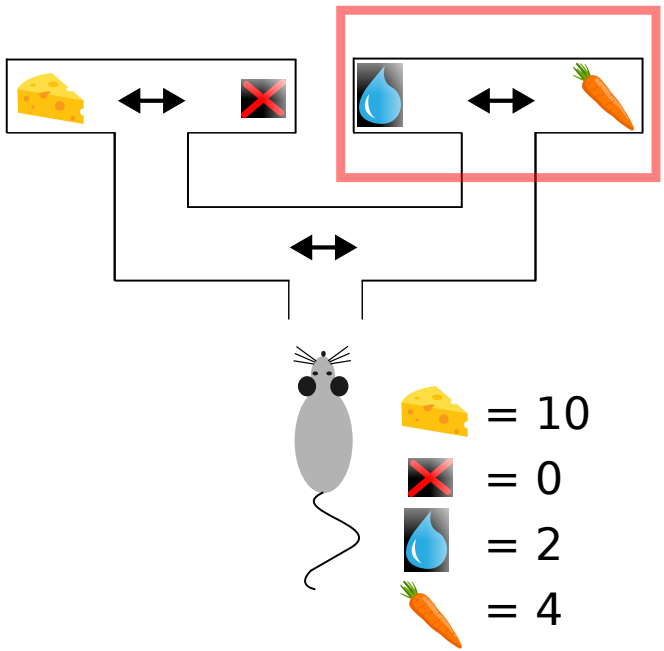


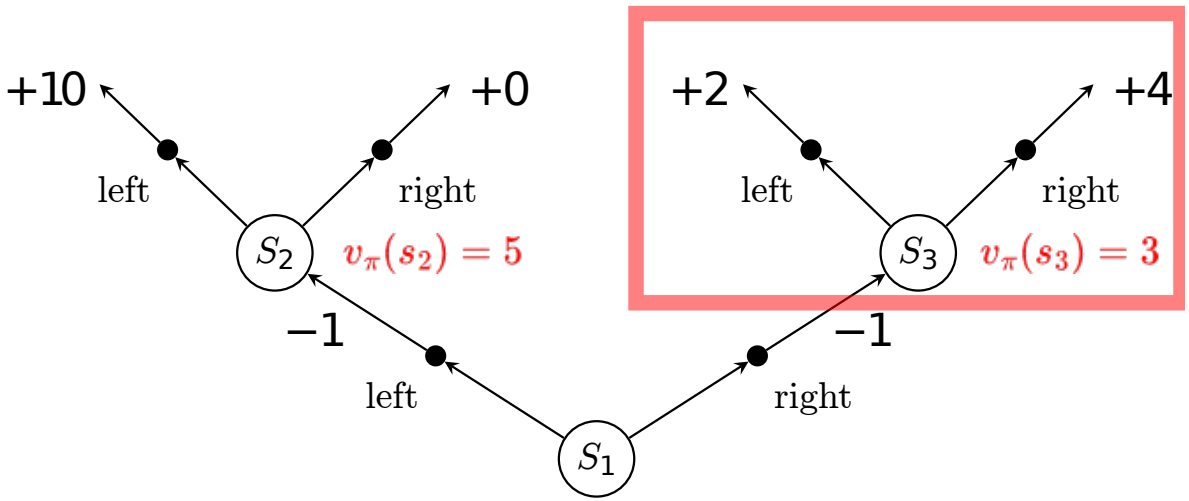


simple example: random policy

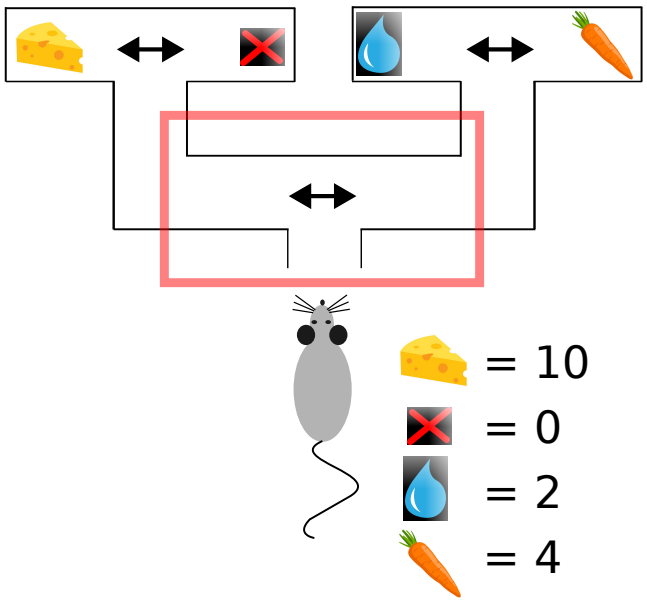


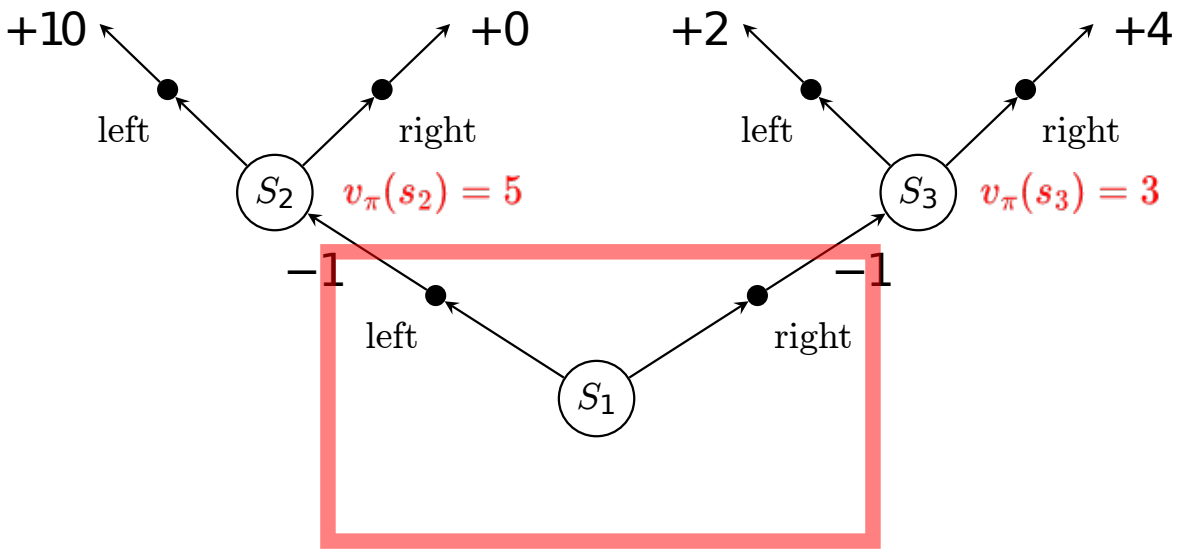


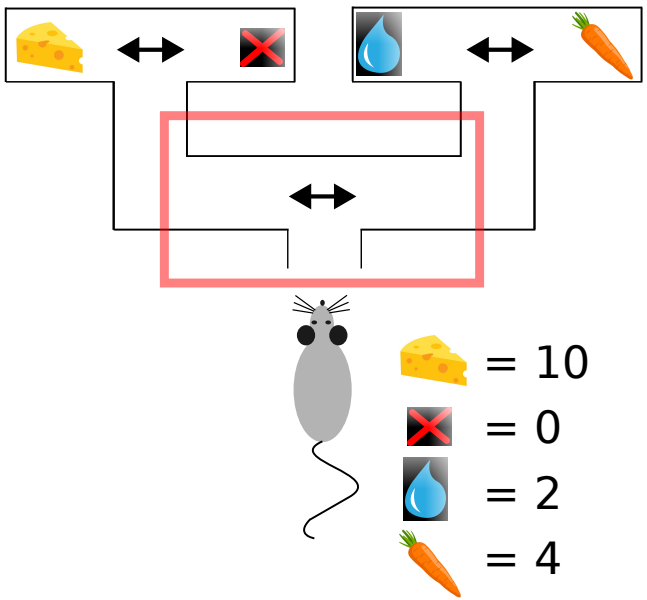


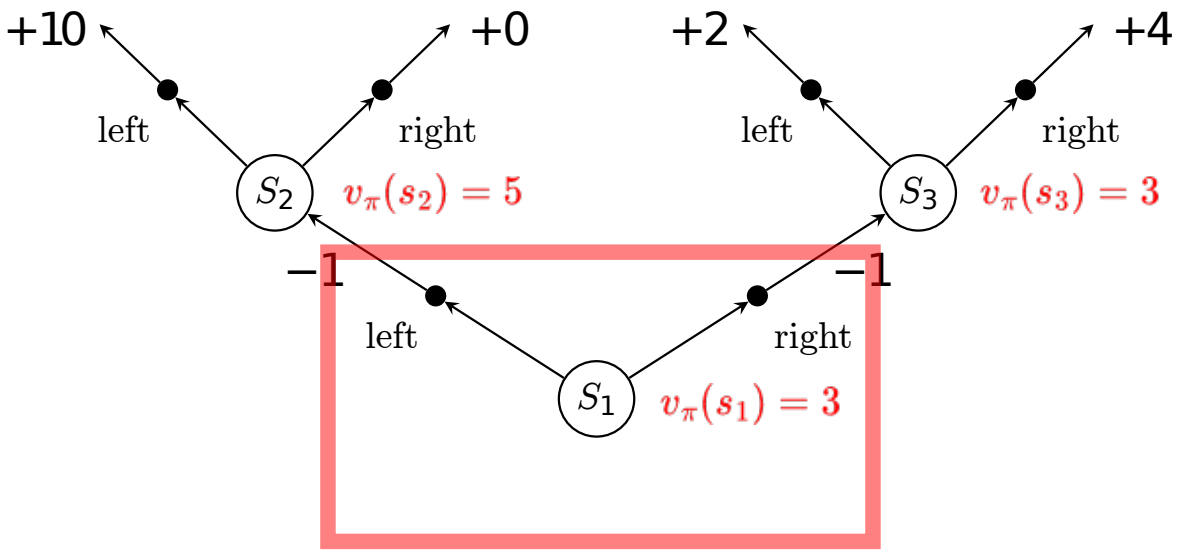


simple example: random policy

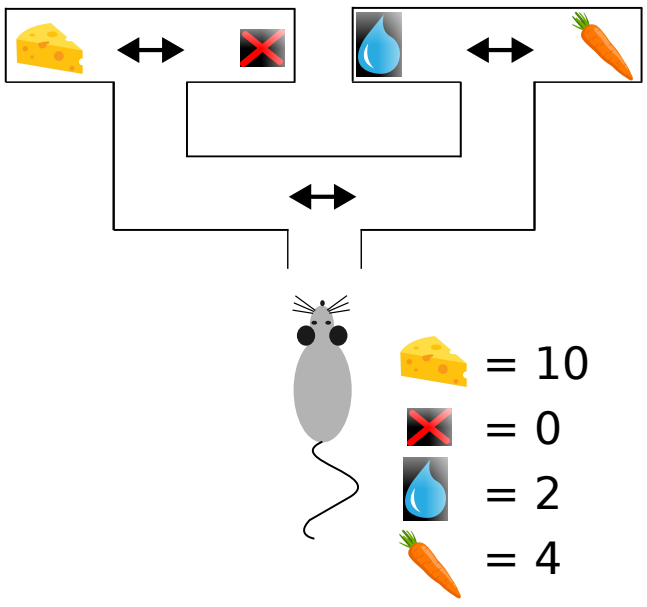


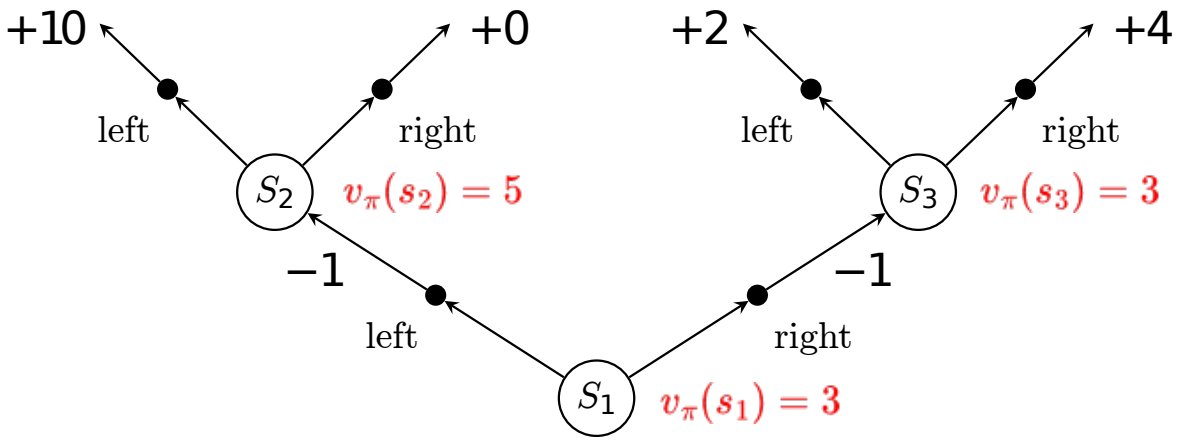






simple example: random policy





'policy evaluation'

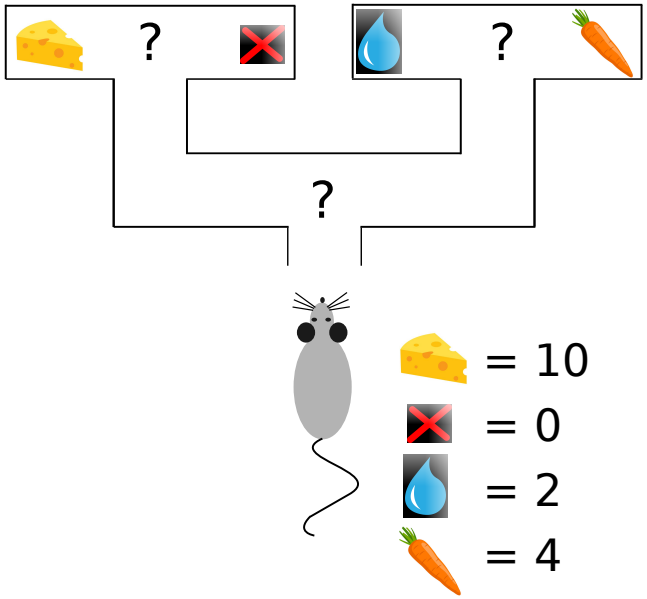
solving it: optimal value function

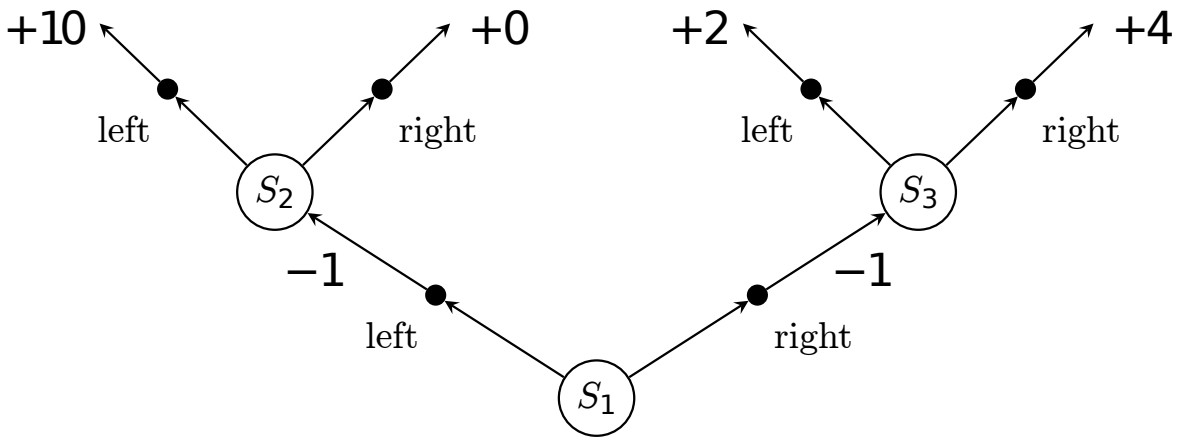
Consistency conditions also true of values under an optimal policy π^*:

$$\begin{aligned} v_*(s) &= \mathbb{E}_{\pi^*}[R_{t+1} + \gamma v_*(s') | S_t = s] \\ &= \max_a \; q_*(s, a) \end{aligned}$$

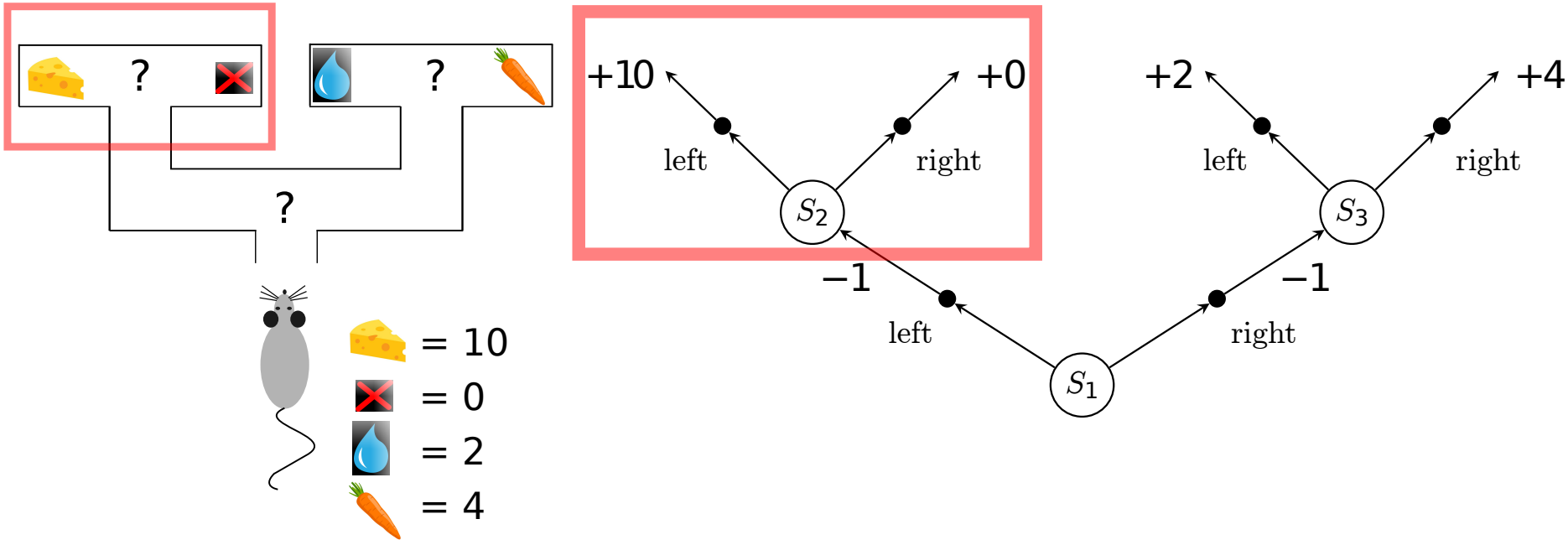
'Bellman (optimality) equations'

simple example: optimal policy





simple example: optimal policy



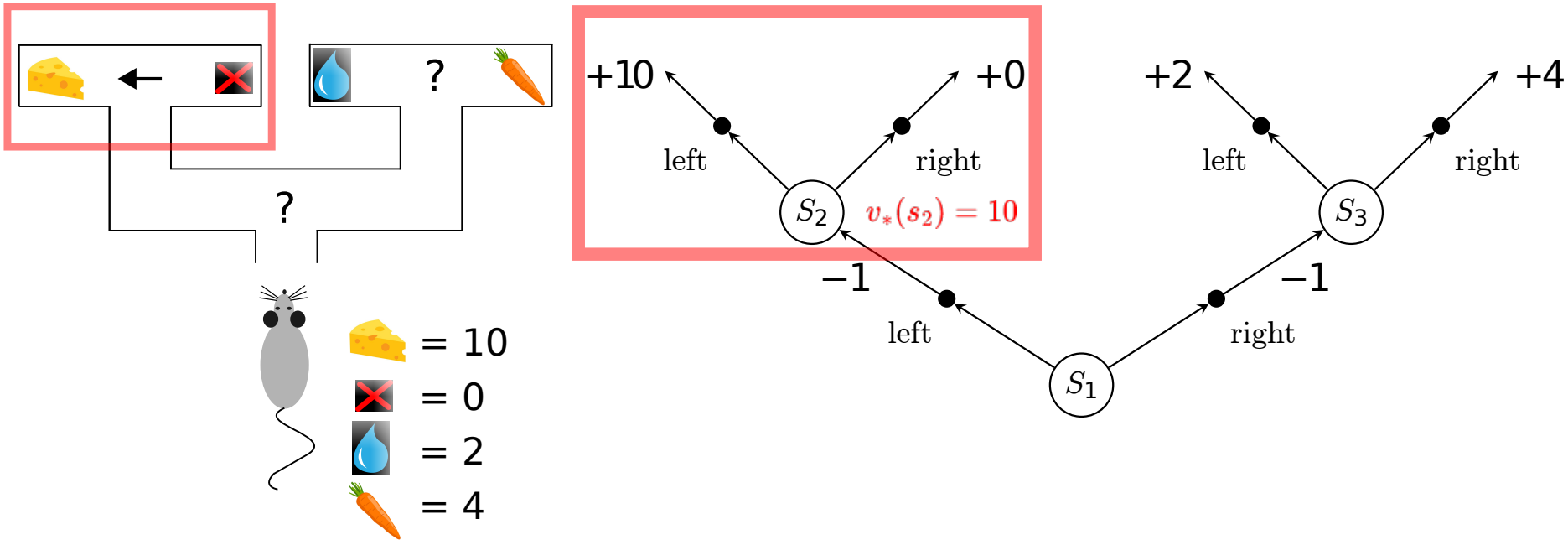
$$q_*(s_2, a = \text{left}) = 10$$

$$q_*(s_2, a = \text{right}) = 0$$

$$v_*(s_2) = \max_a \; q_*(s_2, a)$$

$$= 10$$

simple example: optimal policy



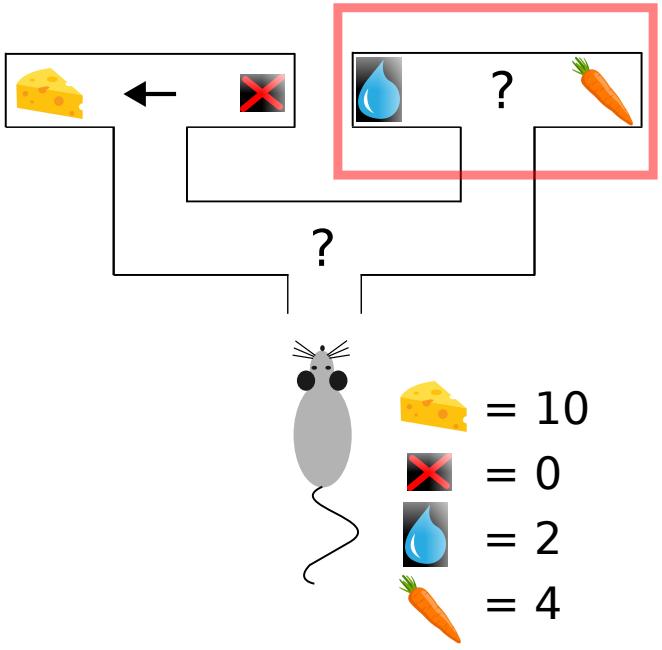
$$q_*(s_2, a = \text{left}) = 10$$

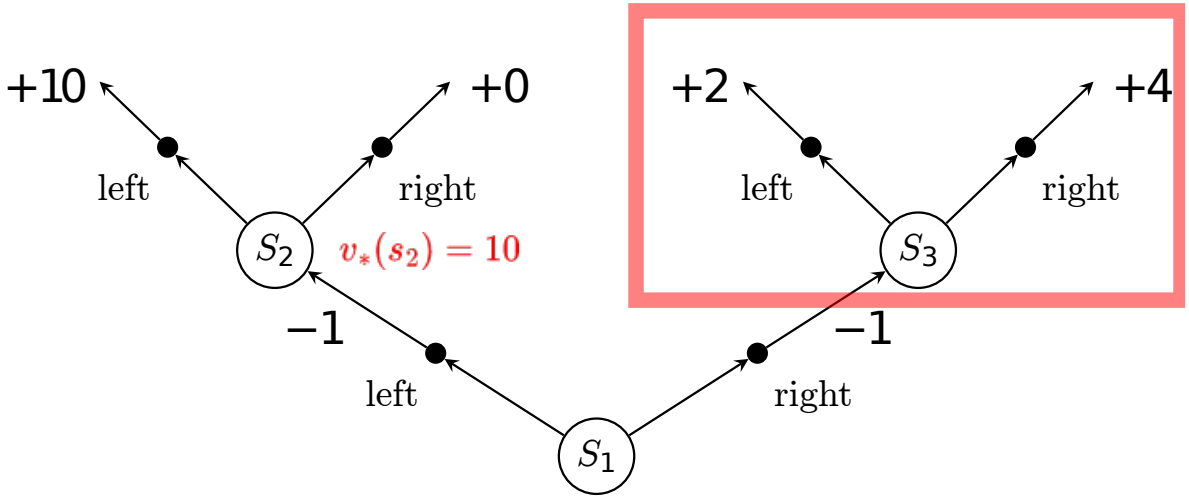
$$q_*(s_2, a = \text{right}) = 0$$

$$v_*(s_2) = \max_a \; q_*(s_2, a)$$

$$= 10$$

simple example: optimal policy





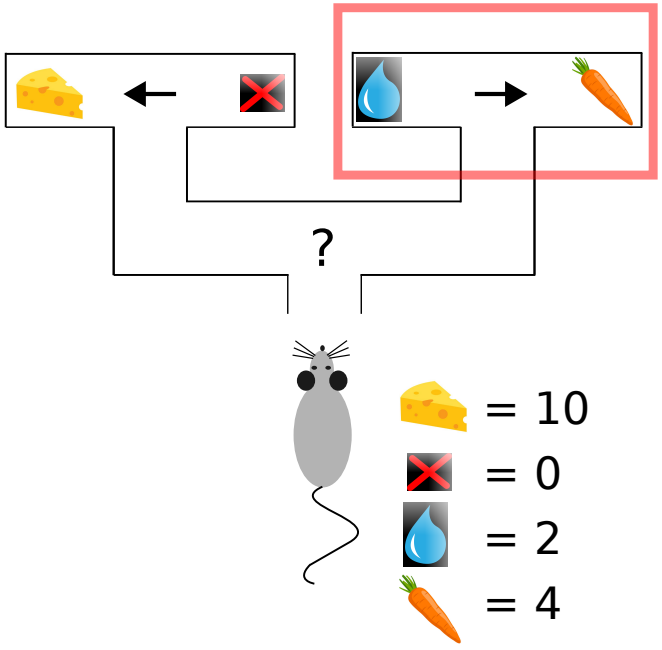
$$q_*(s_3, a = \text{left}) = 2$$

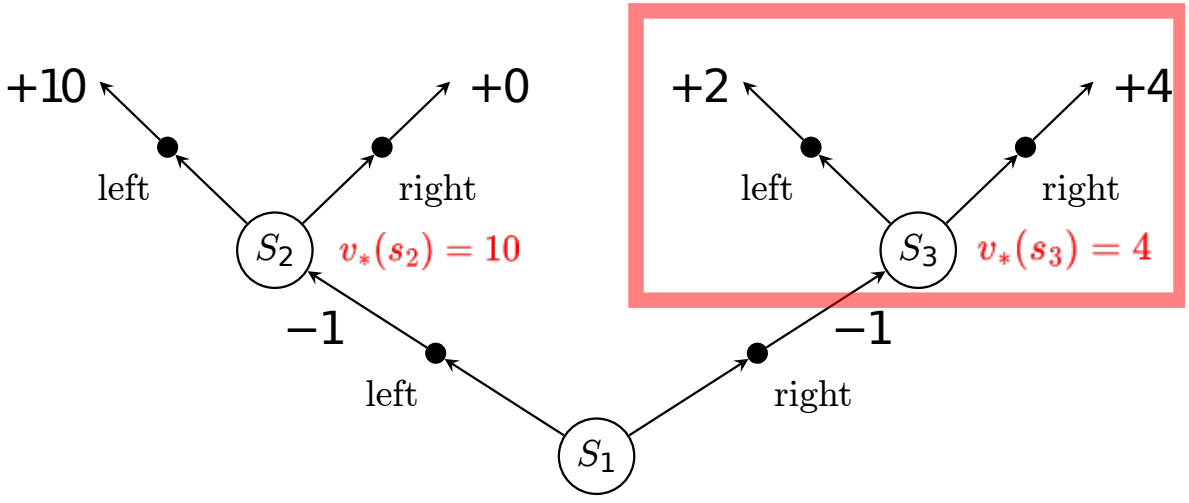
$$q_*(s_3, a = \text{right}) = 4$$

$$v_*(s_3) = \max_a \; q_*(s_3, a)$$

$$= 4$$

simple example: optimal policy





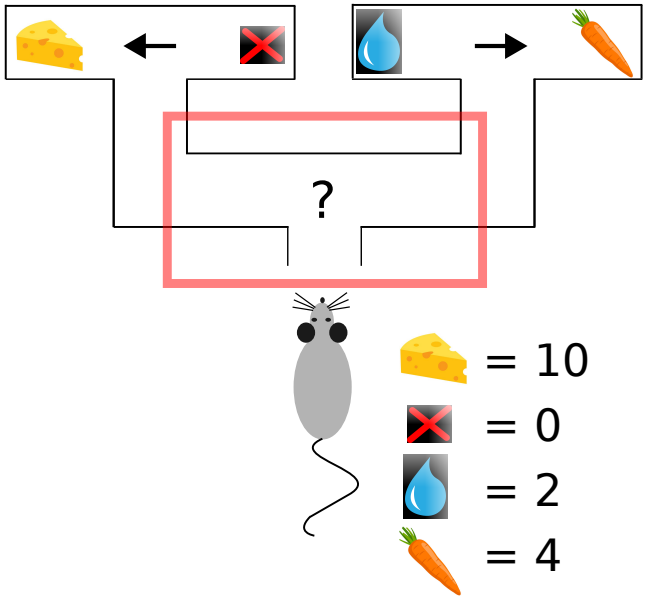
$$q_*(s_3, a = \text{left}) = 2$$

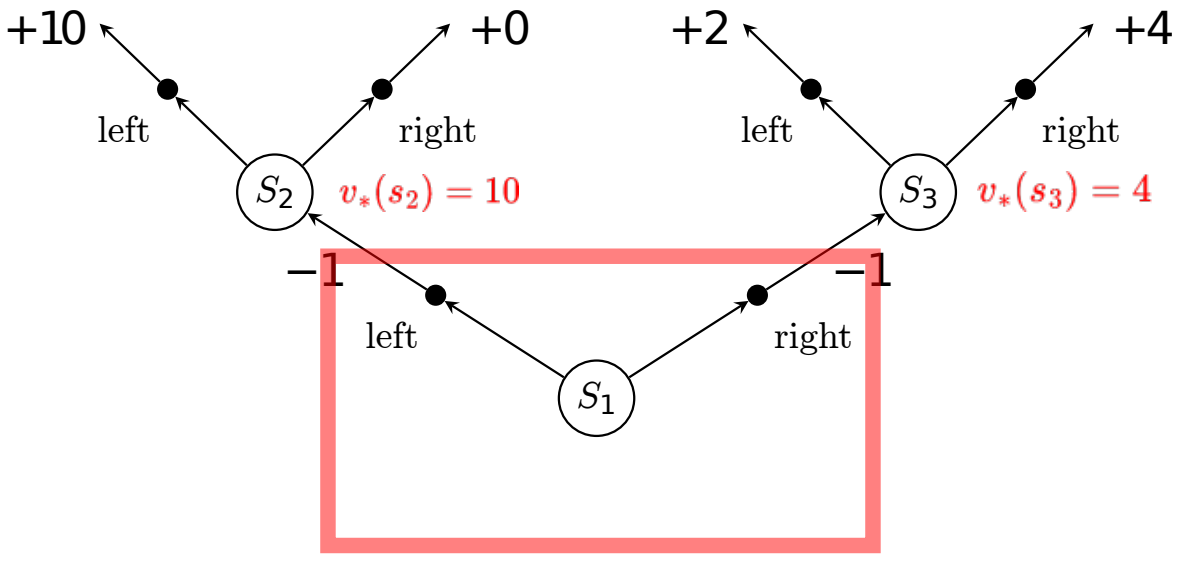
$$q_*(s_3, a = \text{right}) = 4$$

$$v_*(s_3) = \max_a \; q_*(s_3, a)$$

$$= 4$$

simple example: optimal policy





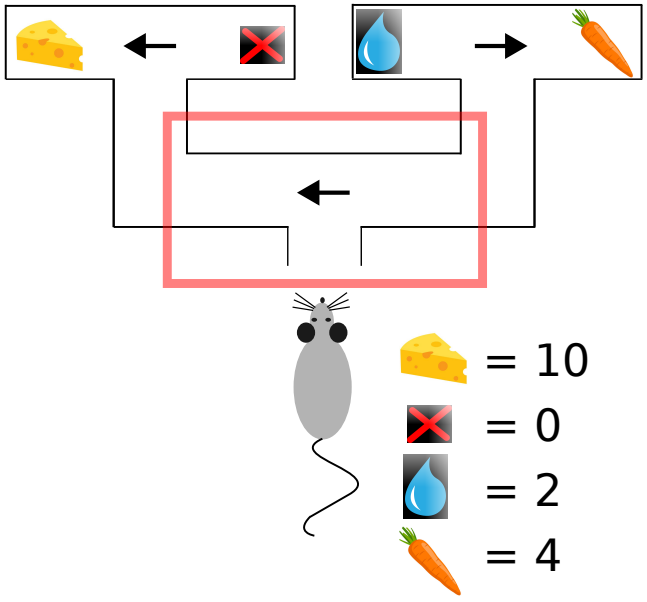
$$q_*(s_1, a = \text{left}) = r + \gamma v_*(s_2) = -1 + 10 = 9$$

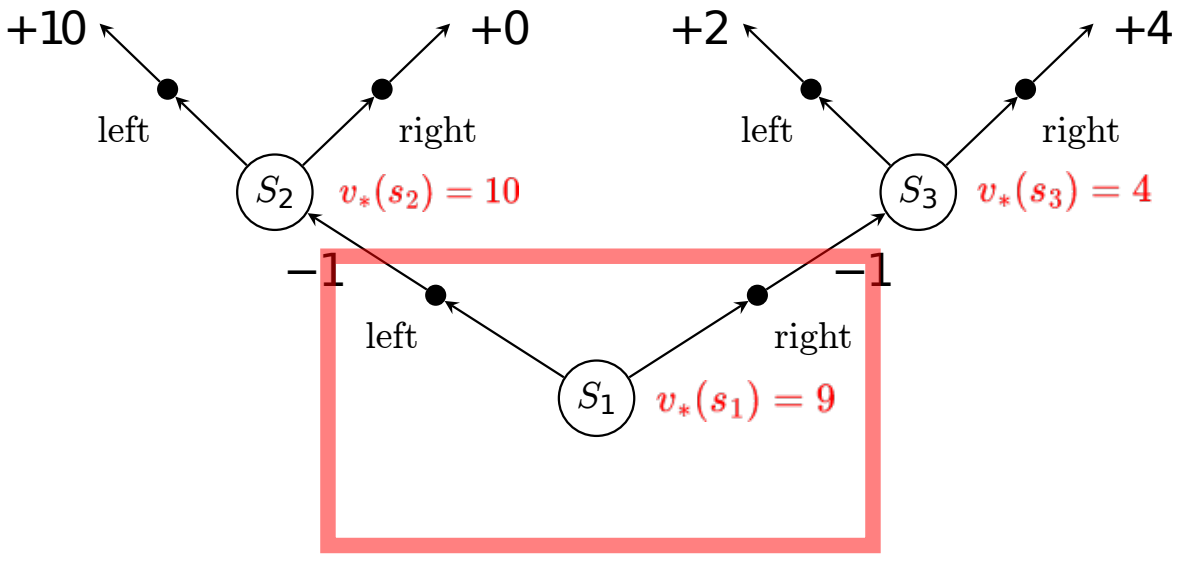
$$q_*(s_1, a = \text{right}) = r + \gamma v_*(s_3) = -1 + 4 = 3$$

$$v_*(s_1) = \max_a \; q_*(s_1, a)$$

$$= 9$$

simple example: optimal policy





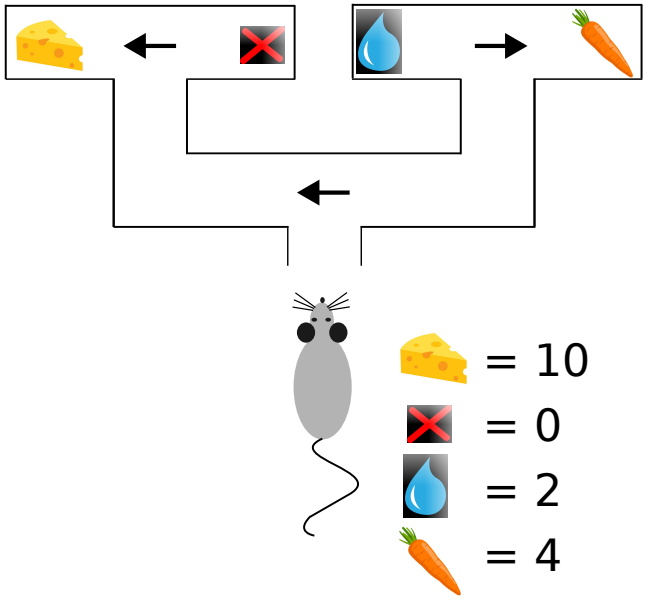
$$q_*(s_1, a = \text{left}) = r + \gamma v_*(s_2) = -1 + 10 = 9$$

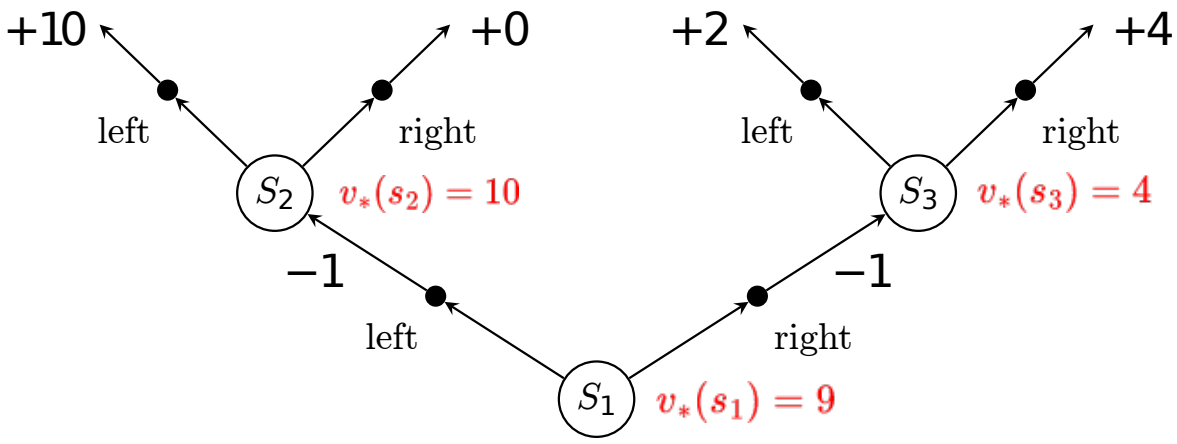
$$q_*(s_1, a = \text{right}) = r + \gamma v_*(s_3) = -1 + 4 = 3$$

$$v_*(s_1) = \max_a \; q_*(s_1, a)$$

$$= 9$$

simple example: optimal policy





'backwards induction'

- but what if there's no 'final stage' (i.e., MDP has **cycles**)?
  $\rightarrow$ **iterative solution methods** ('dynamic programming')
- but what if you don't know the dynamics, p?
  $\rightarrow$ **reinforcement learning methods**

dynamic programming

1. **policy iteration** [see S & B, p.80]
   Iterate between **policy evaluation (E)** and **policy improvement (I)**

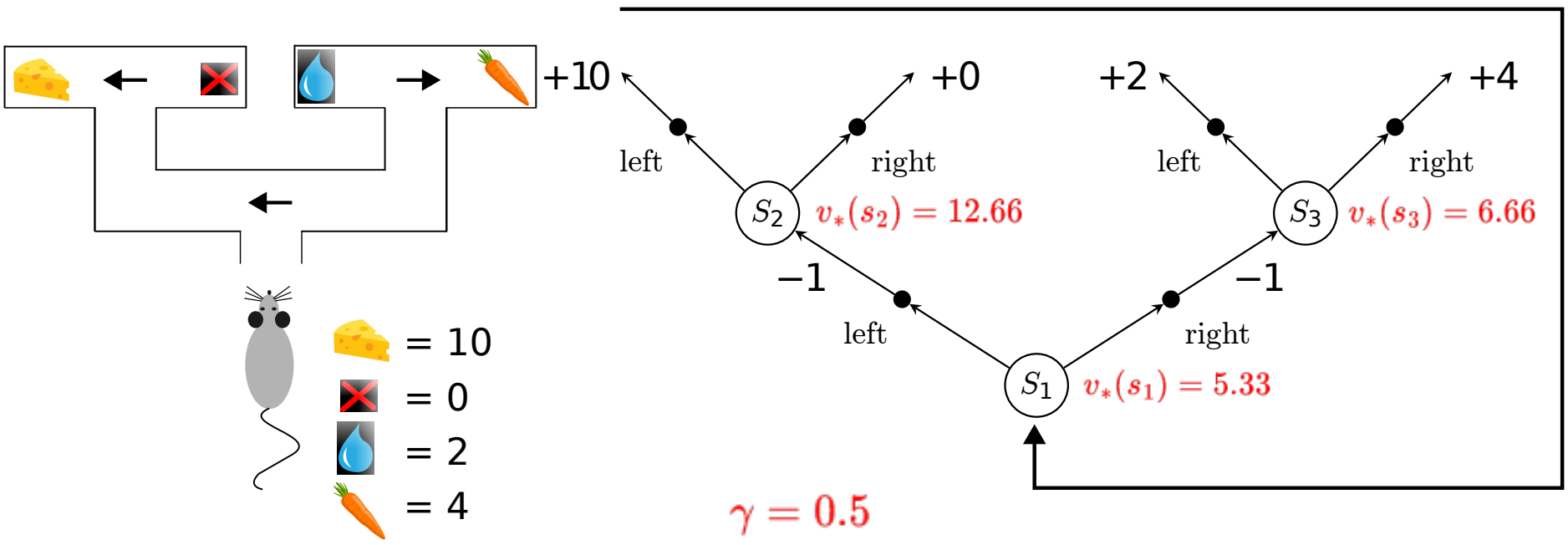
$$\pi_0 \xrightarrow{\text{E}} v_{\pi_0} \xrightarrow{\text{I}} \pi_1 \xrightarrow{\text{E}} v_{\pi_1} \xrightarrow{\text{I}} \pi_2 \xrightarrow{\text{E}} \ldots \xrightarrow{\text{I}} \pi_* \xrightarrow{\text{E}} v_*$$

   E: for each state, $v_\pi(s) = \mathbb{E}_\pi[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s]$
   I: for each state, $\pi'(s) \leftarrow \arg\max_a q_\pi(s, a)$

2. **value iteration** [see S & B, p.83]
   Turn Bellman optimality equation into an update!

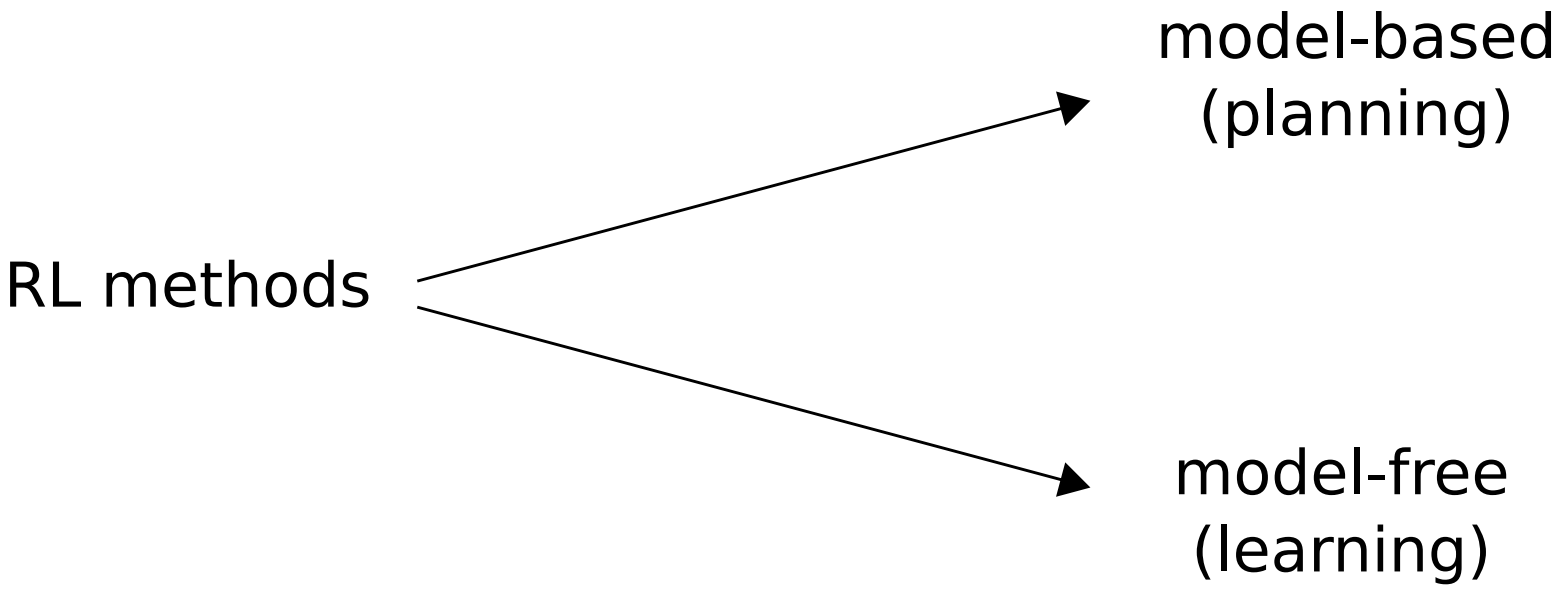
$$v_*(s) = \max_a \mathbb{E}[R_{t+1} + \gamma v_*(S_{t+1}) | S_t = s, A_t = a]$$

(homework?) example

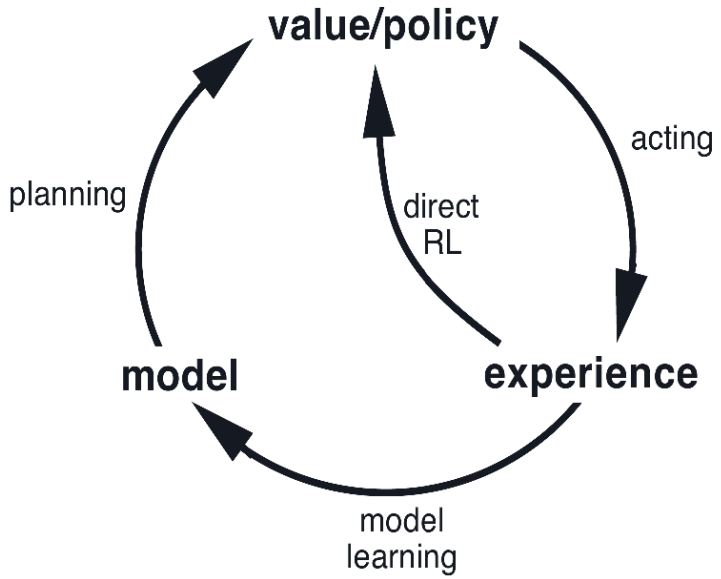


RL methods

. . . all of these [RL] methods can be viewed as attempts to achieve much the same affect as DP [Dynamic Programming], only with less computation and without assuming a perfect model of the environment. (Sutton & Barto, 2018, p.73)



RL methods: integrated planning and learning



(Sutton & Barto, 2018, Ch.8)

- **model-based/indirect**
  - +ve: fuller use of limited experience (model-building & propagation of value) $\rightarrow$ better policy with fewer environmental interactions
  - -ve: possible problems if model is biased
- **model-free/direct**
  - +ve: simpler and not affected by possible model bias ('world is its own best model' – Brooks)
  - -ve: more experience required to achieve a good policy

cf. Daw et al., 2005: 'Uncertainty-based competition between prefrontal and dorsolateral striatal systems for behavioral control'

flexibility of MDPs

MDP framework is abstract and (therefore) flexible:

- **states**
  - may be more or less abstract (e.g., low-level sensory input vs. symbolic description of environment layout)
  - can include both 'external' and 'internal' factors (e.g., memory of past sensations, degree of uncertainty, other states of mind)
- **actions**
  - may be more or less abstract (e.g., low-level motor control vs. 'take train to Zürich')
  - could be entirely mental/computational (e.g., what to think about/pay attention to/remember)

  [see, e.g., Dayan (2012): 'How to set the switches on this thing']

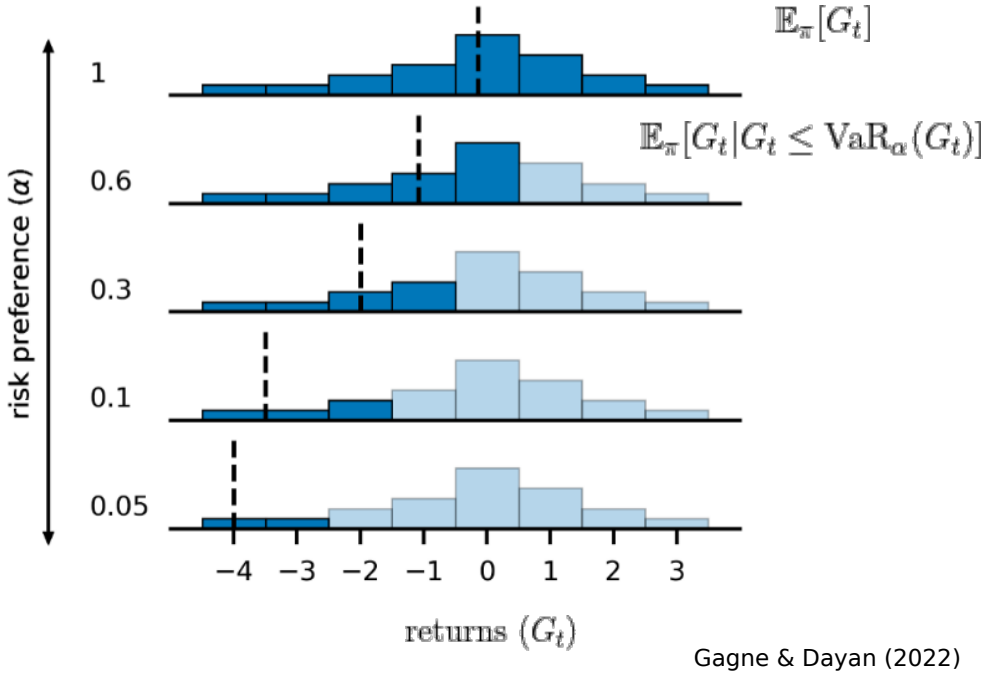
altered decision-making (anything that can go wrong...)

[cf. Huys et al. (2015), 'Decision-theoretic psychiatry']

- **'solving the wrong problem'**
  - altered rewards $\mathcal{R}$
    - 'flattening' for +ve outcomes (anhedonia?)
    - skewed rewards for addictive substances
    - discounting $\gamma \to 0$ as route to impulsivity
  - representation of states $\mathcal{S}$ & actions $\mathcal{A}$
    - consequences for (under-/over-)generalization
  - incorrect dynamics function/model p
    - e.g. more dangerous/less controllable than reality
    - can be difficult to correct – we are **active agents**
- **'solving the right problem, but poorly'**
  - curse of dimensionality + limits on computation & memory $\to$ approximations & heuristics
    - balancing model-based (goal-directed) and model-free (habitual) control
    - balancing instrumental (MF/MB) and Pavlovian (evolutionarily pre-wired) control

extension: risk-sensitive MDPs

- formalizing risk preferences using conditional value-at-risk (CVaR)



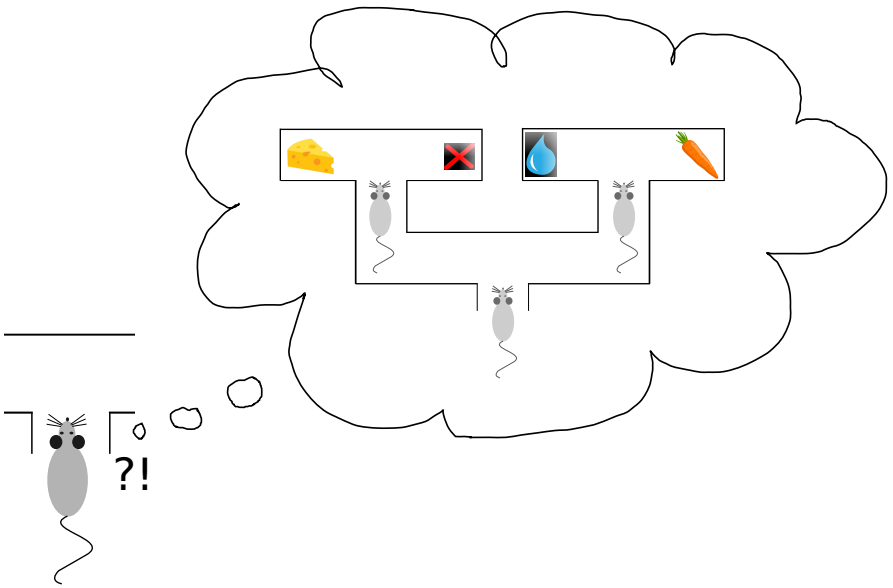
Gagne & Dayan (2022)

- can formulate Bellman-like equations for risk-sensitive values in terms of (pessimistically) *distorted* dynamics $\widetilde{p}$

(see Gagne & Dayan, 2022: 'Peril, prudence and planning as risk, avoidance and worry')

extension: partial observability (POMDPs)

- in most cases, sensory input only provides *partial* information about environment state; **observations** $O_1, O_2, \ldots$
- current observation may not be sufficient to predict the future (i.e., **non-Markov**) — problematic for solution methods for MDPs



- possible solutions:
  - seek a (compact) summary of *history* of observations (not just current observation) that is Markov
  - maintain a *distribution* over latent states (**'belief' MDP**)

summary

- the Markov decision process (MDP) — and its extensions — provides a very flexible formal framework to rigorously study decision problems
- Dynamic programming (DP) algorithms, based on Bellman equations, can be used (in principle) to find optimal solutions — and are the foundation for approximate methods (e.g., RL) that aim at these solutions
- part of a framework for thinking about how decision-making may be altered in psychiatric populations

Questions?