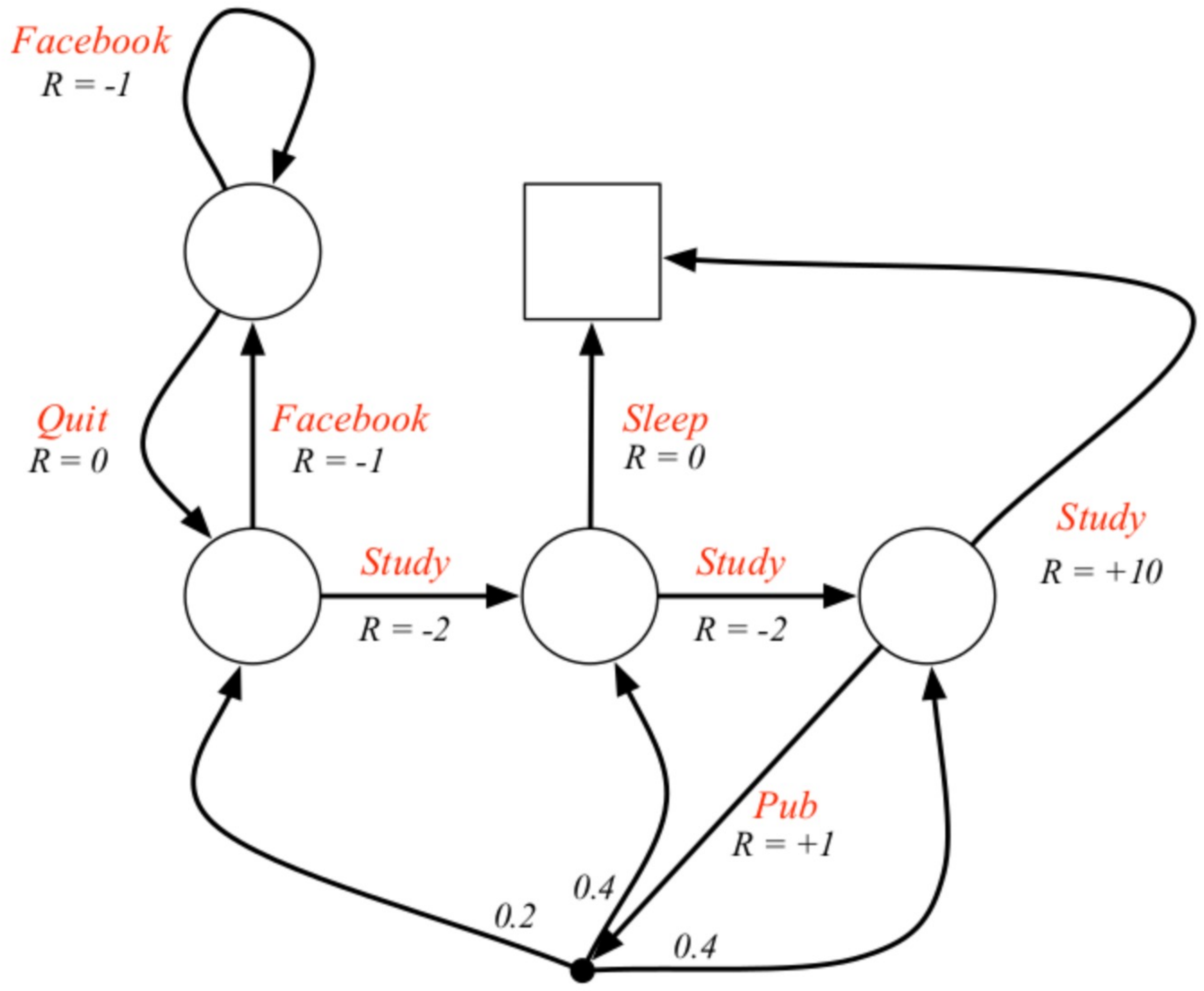
CSCI 3202: Intro to Artificial Intelligence
Lecture 32: Markov Decision Processes

Rhonda Hoenigman
Department of Computer Science



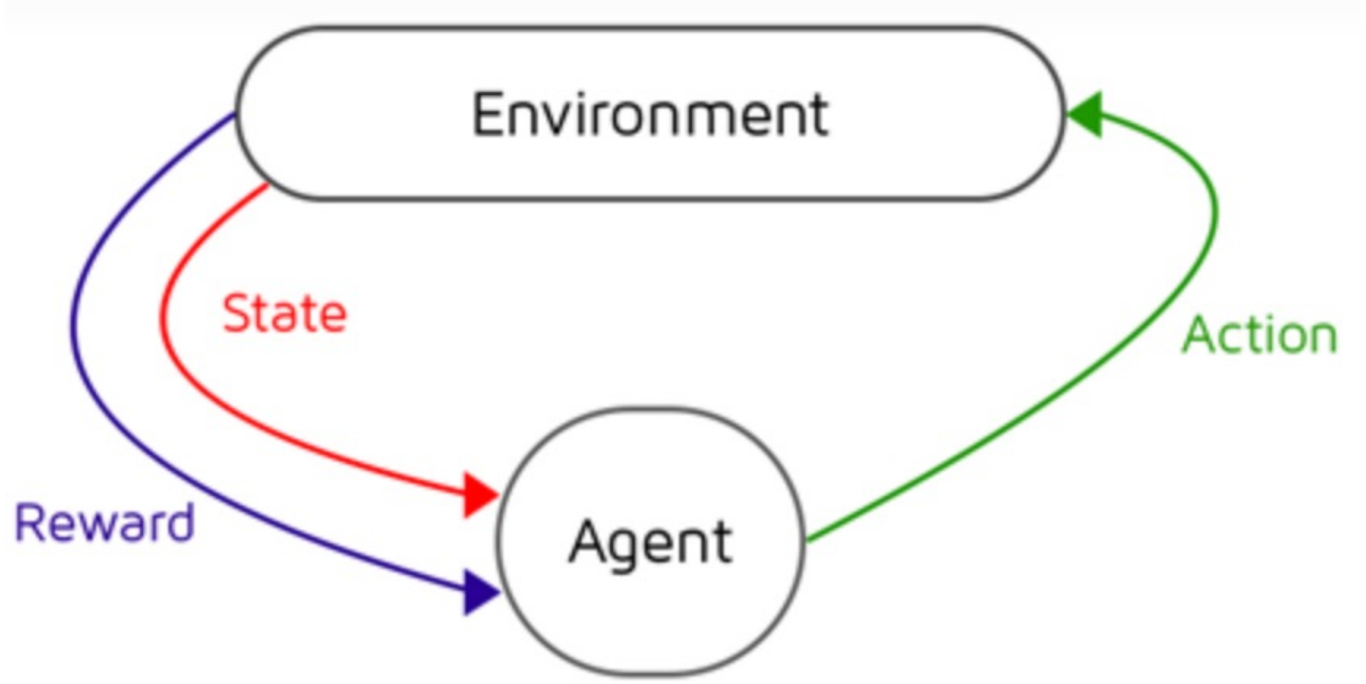
Student MDP with actions

Markov Decision Process – Overview

A Markov Decision Process (MDP): Markov decision processes (MDP) are a framework for sequential decision making. At each step, the agent chooses an action from a set of possible actions.

- Sequential decision problem
- Fully observable, stochastic environment
- Markovian transition model
- Additive reward structure

Used frequently for:
- Inventory management
- Routing/logistics
- Games
- Planning under uncertainty

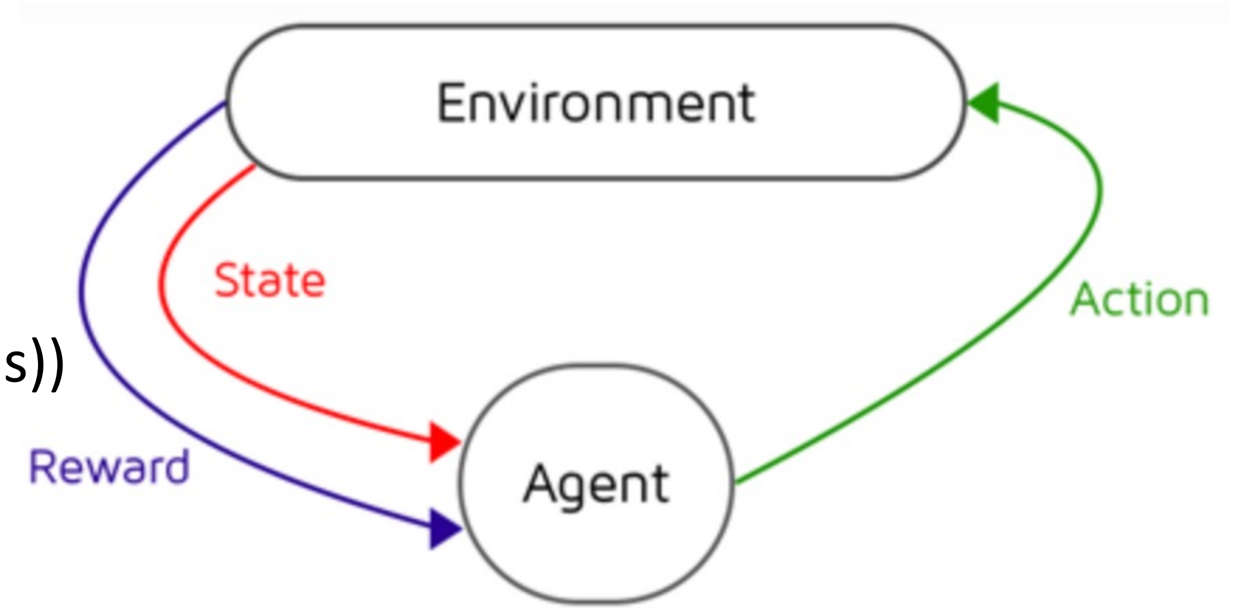


Markov Decision Process – Overview

A Markov Decision Process (MDP): Markov decision processes (MDP) are a framework for sequential decision making. At each step, the agent chooses an action from a set of possible actions.

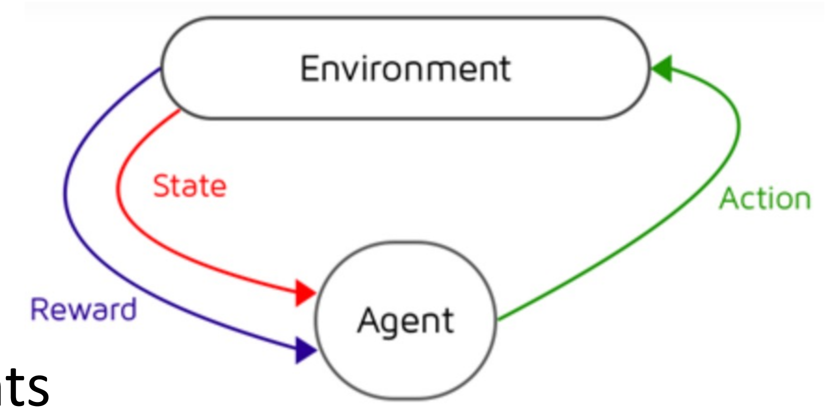
Requires:

- States (call them s, with initial s_0)
- Actions available in each state (Actions (s))
- Transition model (P(s’ | s, a))
- Reward function R(s) (or R(s, a, s’))



Markov Decision Process – Overview

A Markov Decision Process (MDP) objective: In an MDP, we are looking for an in each state that solves the problem and maximize the agent’s expected utility.



Utility of state:

- Real number that captures how well the state represents agent’s goals

- Tell us how good the state is

- Based on reward structure and likelihood of state

Markov Decision Process

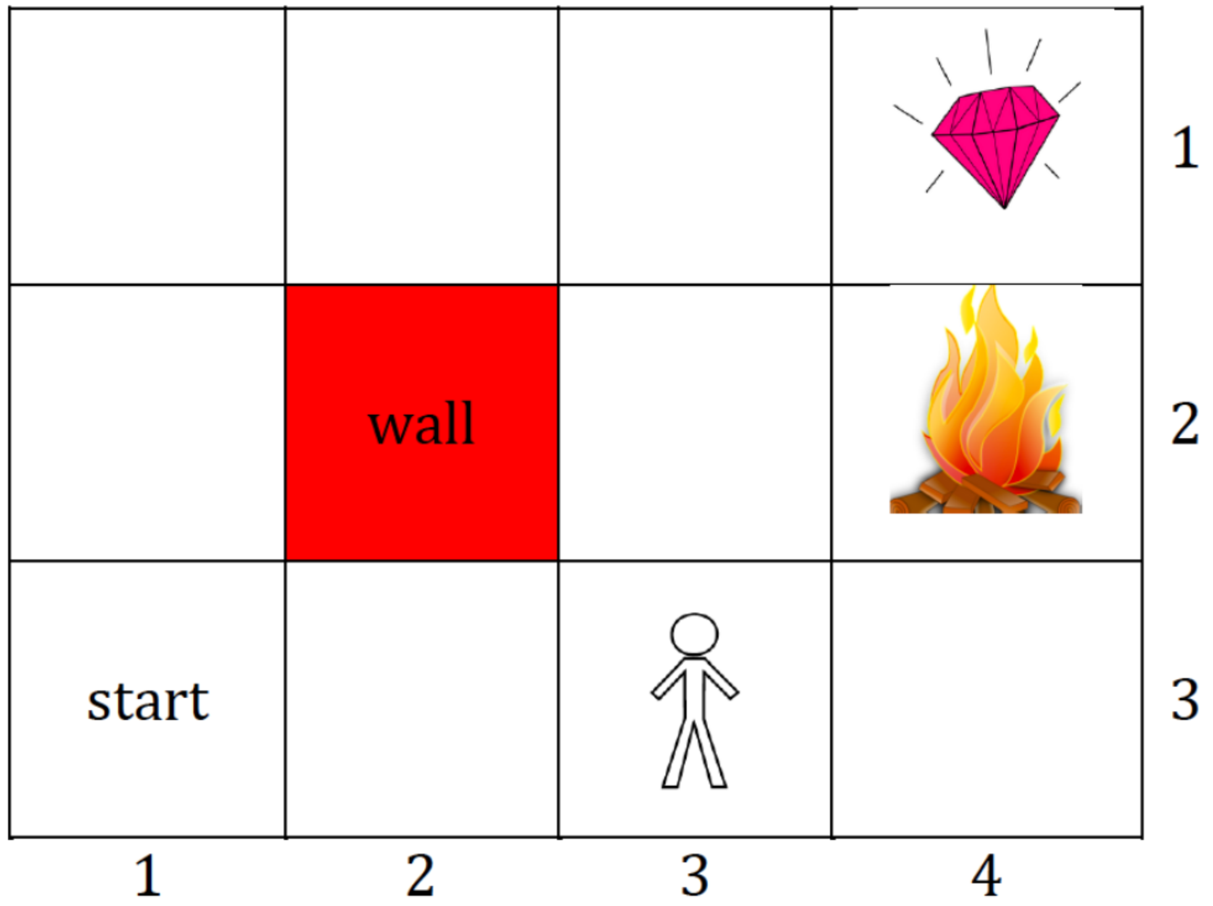
Example: Move agent from “start” to the diamond without falling into the fire pit.

- Diamond has reward of +1.
- Firepit has reward of -1.

- With certainty in actions, the solution is trivial.

- Agent successful 80% of time. 10% of time goes left, 10% of time goes right.

Eg. Assume the agent wants to go north. 80% of the time the action is successful, 10% of the time the agent goes east and 10% of the time the agent goes west. If agent goes into a wall, the agent just bounces off and stays in the same location.



Markov Decision Process – rewards and horizon

<u>Example</u>: Move agent from “start” to the diamond without falling into the fire pit.

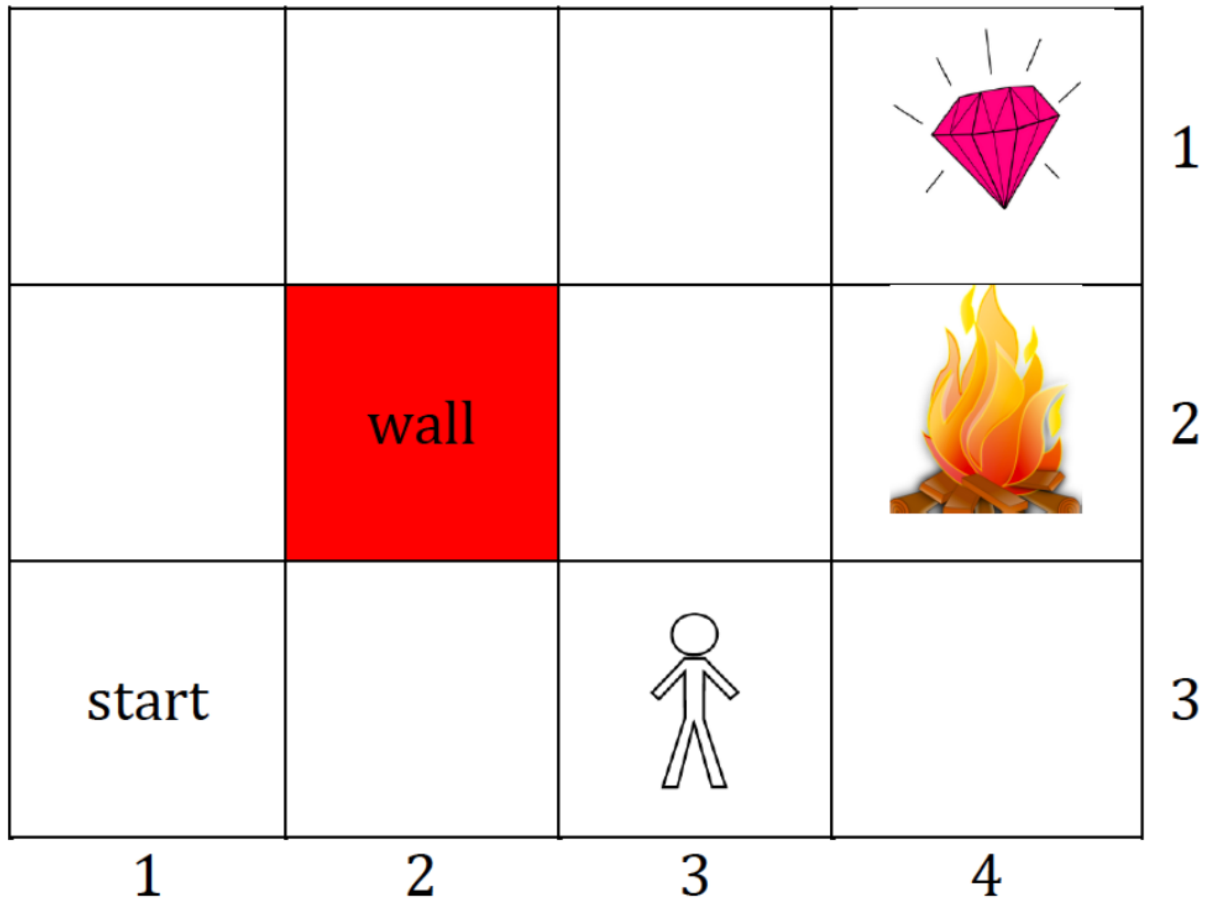
Rewards:

- Diamond has utility of +1.
- Firepit has utility of -1.
- Small living reward, either +/-, that is the utility of the state.

Horizon:

Infinite: No time limit on the problem, optimal action doesn’t depend on time. Rewards can still be discounted. Solution might not be found with additive rewards

Finite: There is a fixed time N after which nothing matters, e.g. the game ends. If we were looking at additive rewards, agent needs to move towards goal.

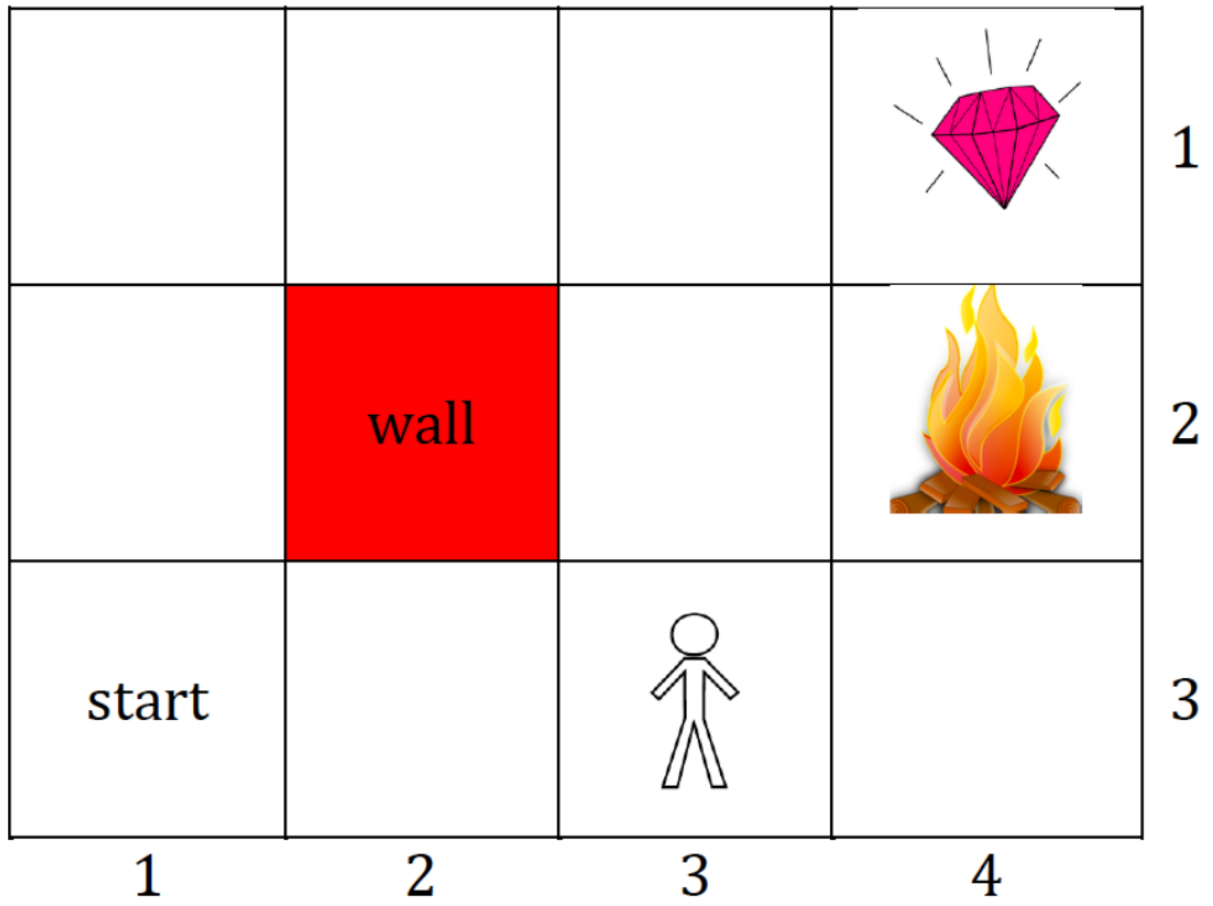


Markov Decision Process – solution

A **solution** for an MDP specifies what the agent should do for any state that the agent might reach. It's not a fixed sequence of actions, but rather an action for a state.

Policy – solution to an MDP, typically denoted as π, where π(s) is the action recommended by policy π in state s.

Optimal policy - has highest expected utility. In calculating the optimal policy, we want to maximize expected utility.

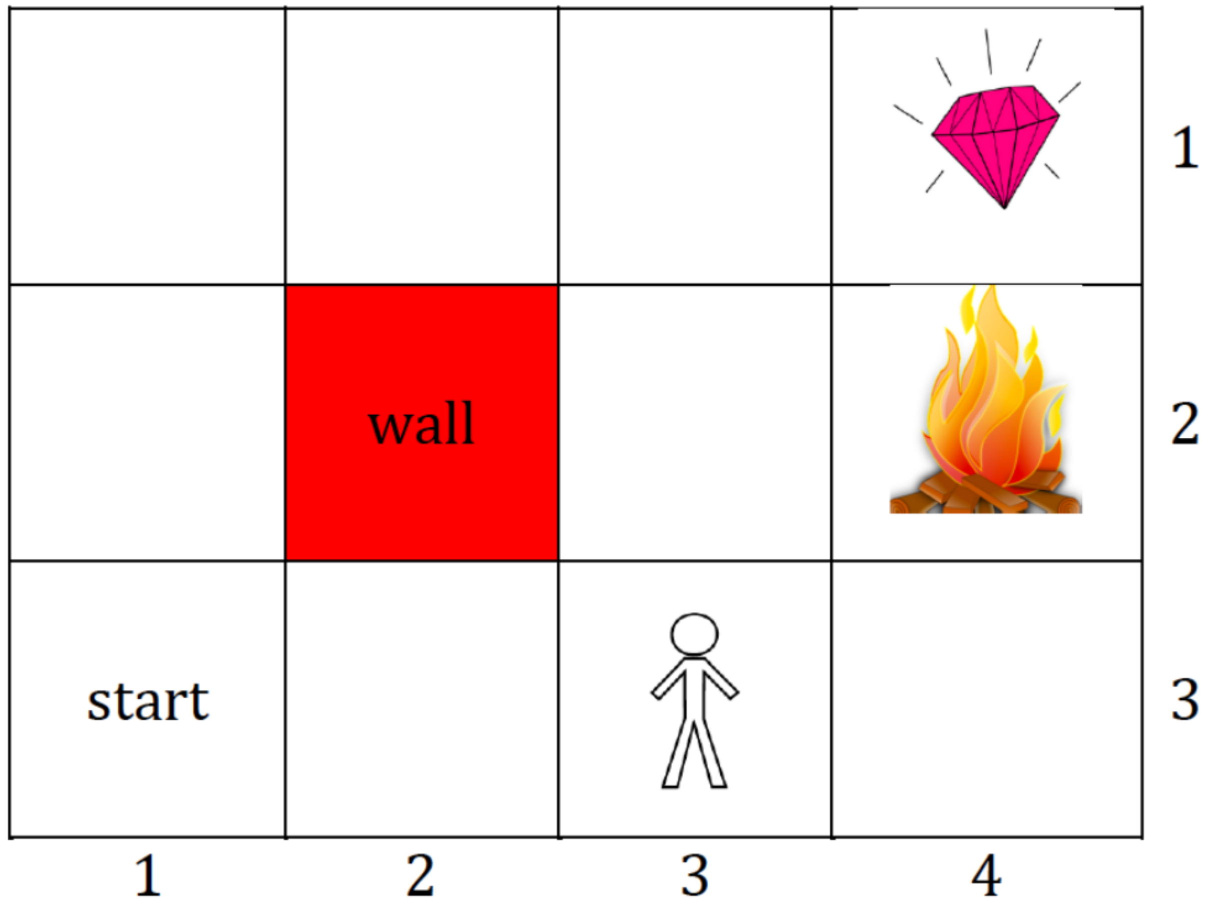


Markov Decision Process – solution

A **solution** for an MDP specifies what the agent should do for any state that the agent might reach. It's not a fixed sequence of actions, but rather an action for a state.

Policy – solution to an MDP, typically denoted as π, where π(s) is the action recommended by policy π in state s.

Optimal policy - has highest expected utility. In calculating the optimal policy, we want to maximize expected utility.



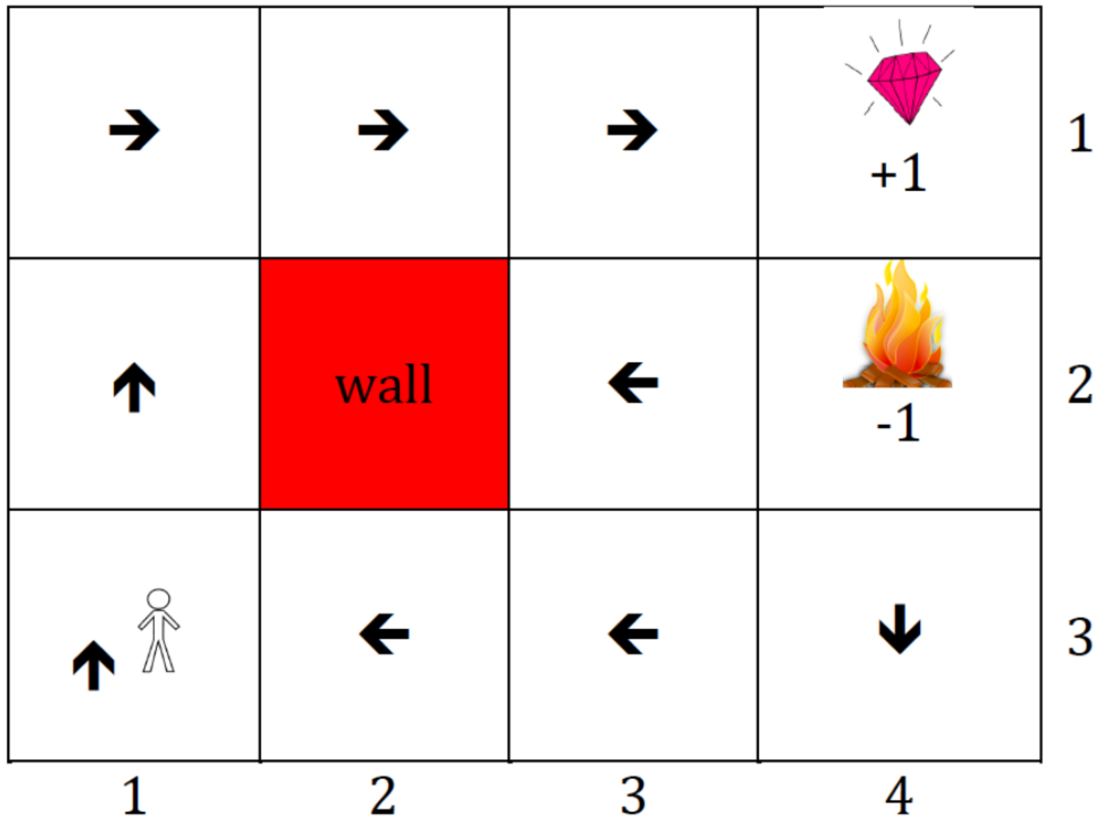
Markov Decision Process – examples of optimal policies

Environment:

- 80% of time action is successful, and 20% of time action is unsuccessful, goes left 10% and right 10%.

- R(s) = -0.01 - Slightly negative living reward.
- On each iteration, the living reward is deducted from the agent's utility.

The optimal policy involves avoiding the fire pit at all costs. In the square next to the pit, run into the wall to avoid the risk of accidentally going into the pit. The living reward deduction is so small that the agent has time to hit the wall over and over.

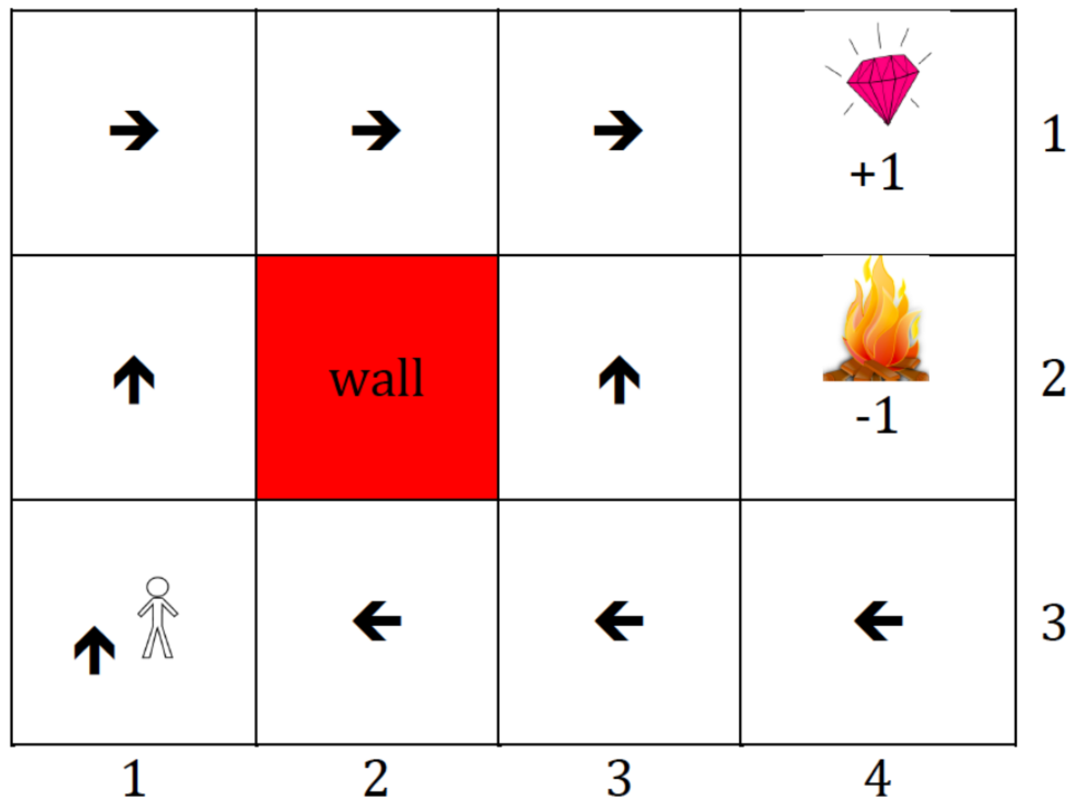


Markov Decision Process – examples of optimal policies

Environment:

- 80% of time action is successful, and 20% of time action is unsuccessful, goes left 10% and right 10%.
- R(s) = -0.03

It is no longer the best policy to run into the wall. The increased risk of dying in the fire is worth it to speed up the path to the diamond.

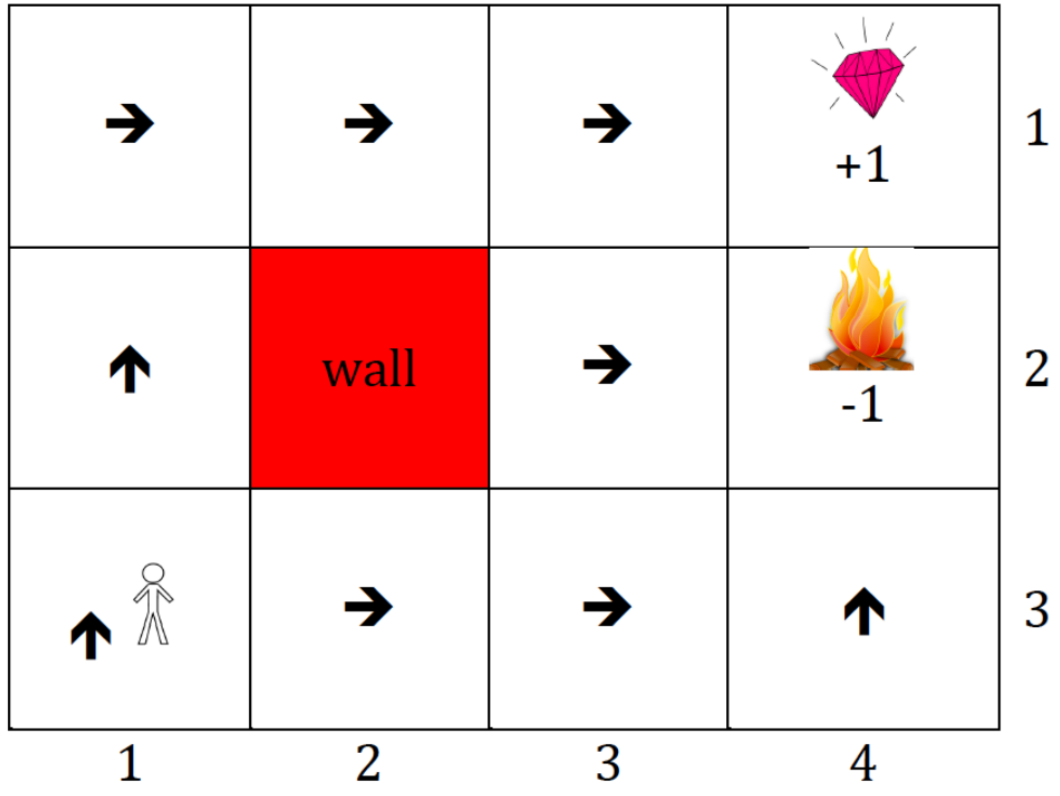


Markov Decision Process – examples of optimal policies

Environment:

- 80% of time action is successful, and 20% of time action is unsuccessful, goes left 10% and right 10%.
- R(s) = -2.0

Jumping into the fire pit is preferable to living.



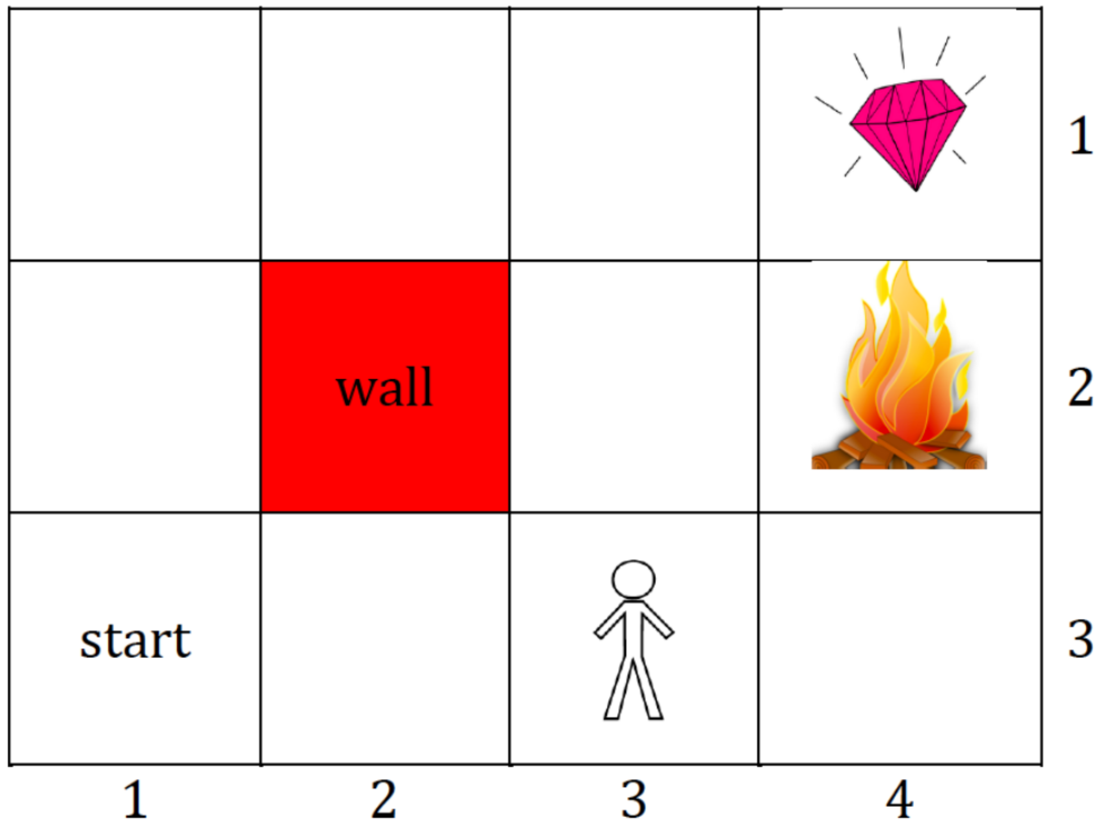
Markov Decision Process

Utility - long-term gain

- Depends on entire sequence of states

$$[s_0, s_1, \ldots, s_{50}, s_{51}, \ldots]$$

- Rough definition:
  Utility = sum of rewards along the sequence

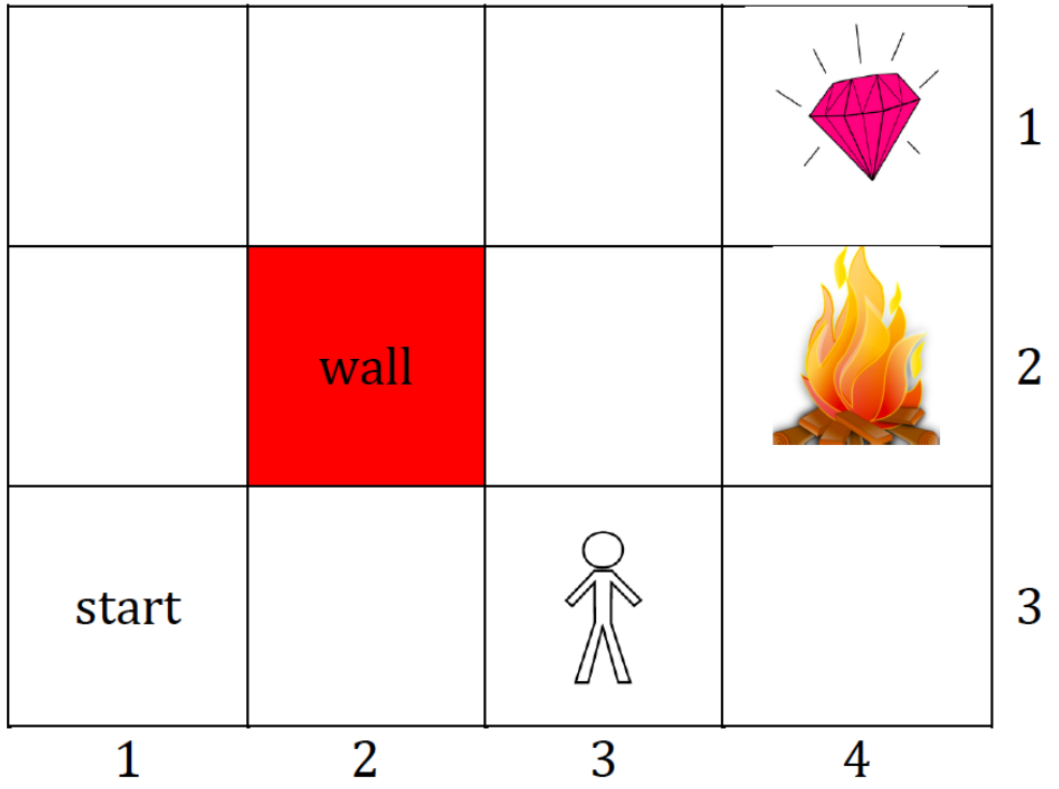


Markov Decision Process

Example: How can we figure out an optimal policy starting from some state s?

- Expected utility under policy π is:

$$U^{\pi}(s) = E\left[\sum_{t=0}^{\infty} \gamma^t R(S_t)\right]$$



Markov Decision Process

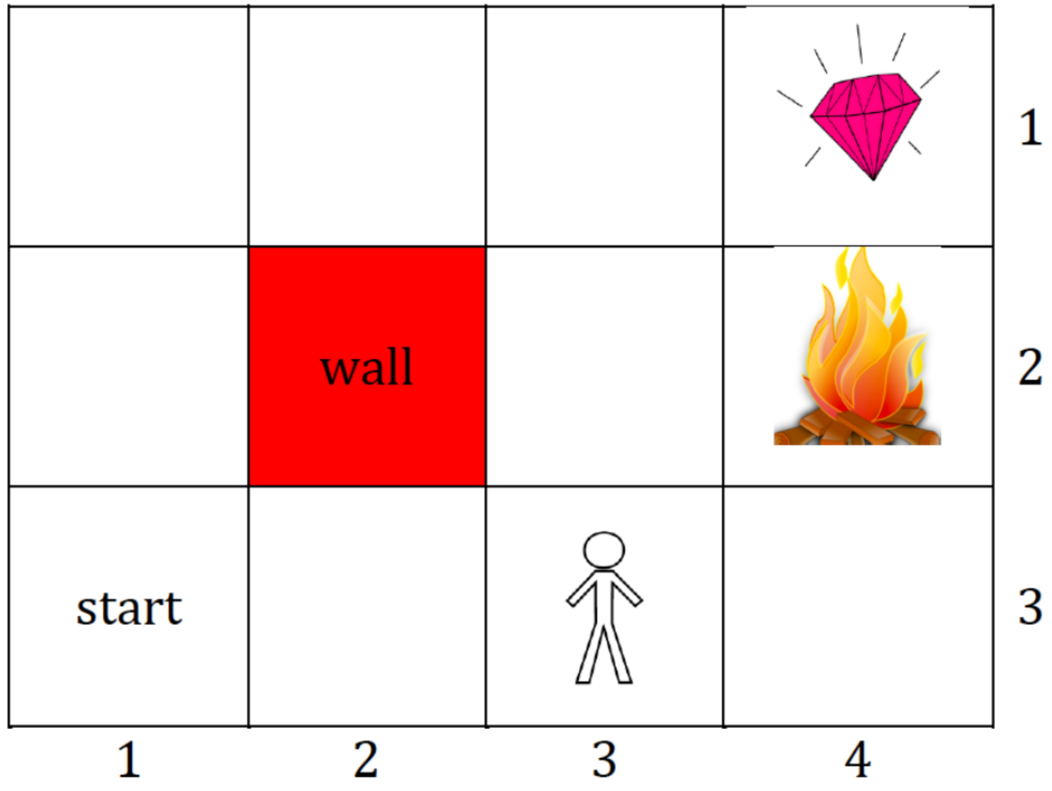
Maximize expected discounted utility

- leads to optimal policy starting from s

$$\pi_s^* = \arg\max_{\pi} U^{\pi}(s)$$

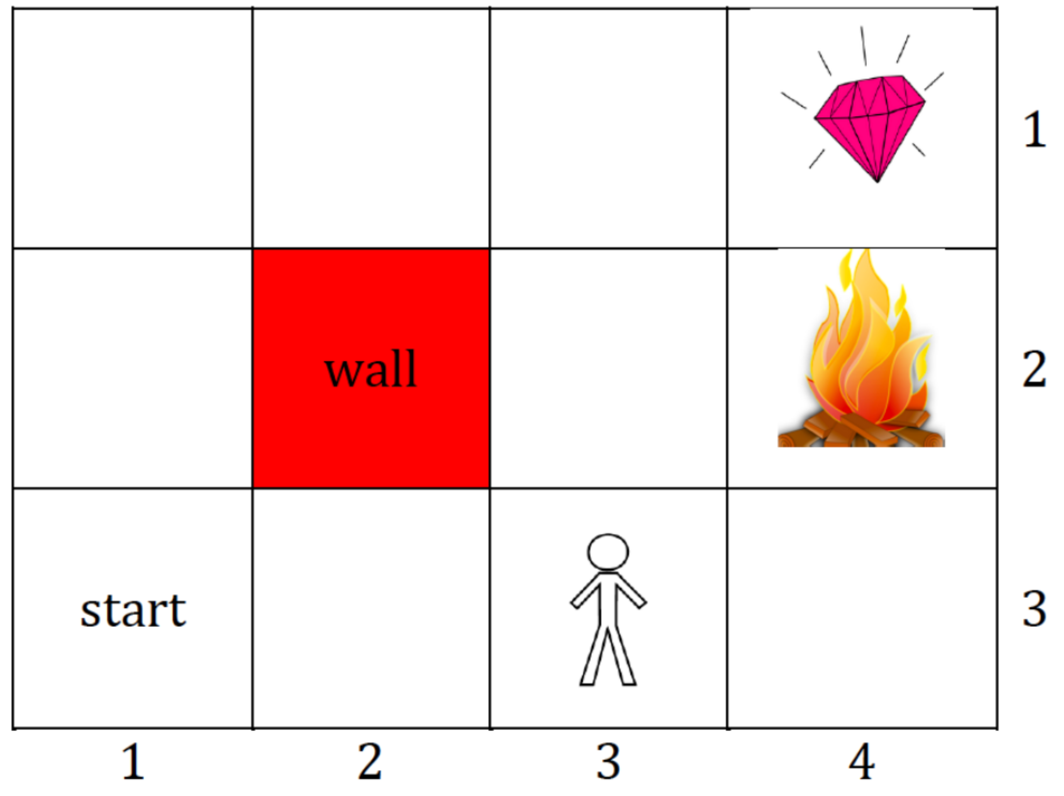
- But policy recommends action for any state, not just one. So we call the optimal policy: π^*

- True utility of a state is $U^{\pi^*}(s)$ - expected discounted sum of rewards if the agent executes an optimal policy from s
  - write this as $U(s)$

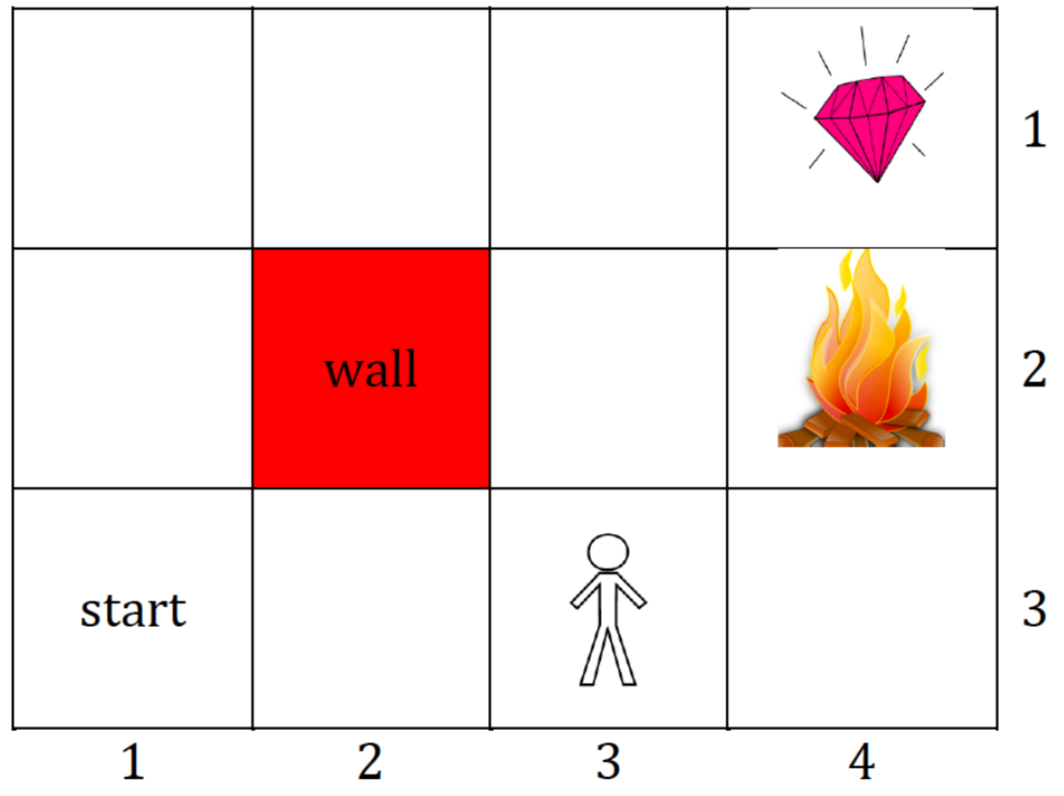


Markov Decision Process

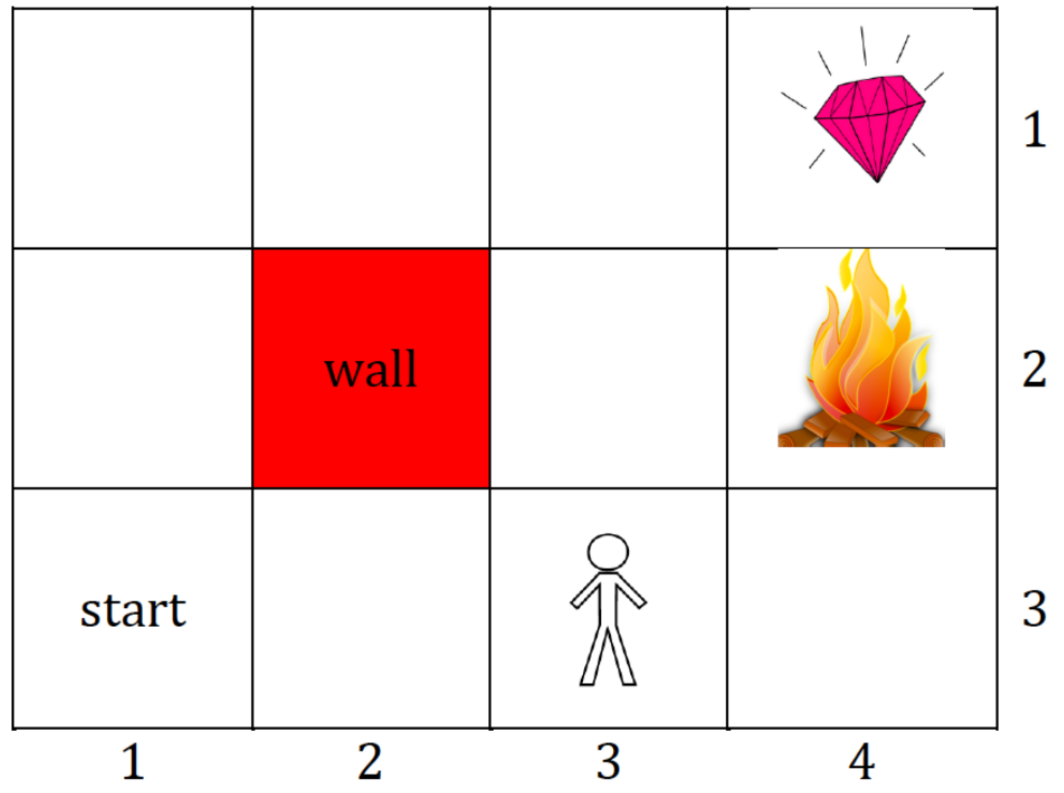
If we know $U(s)$, pick actions to maximize its expected value!



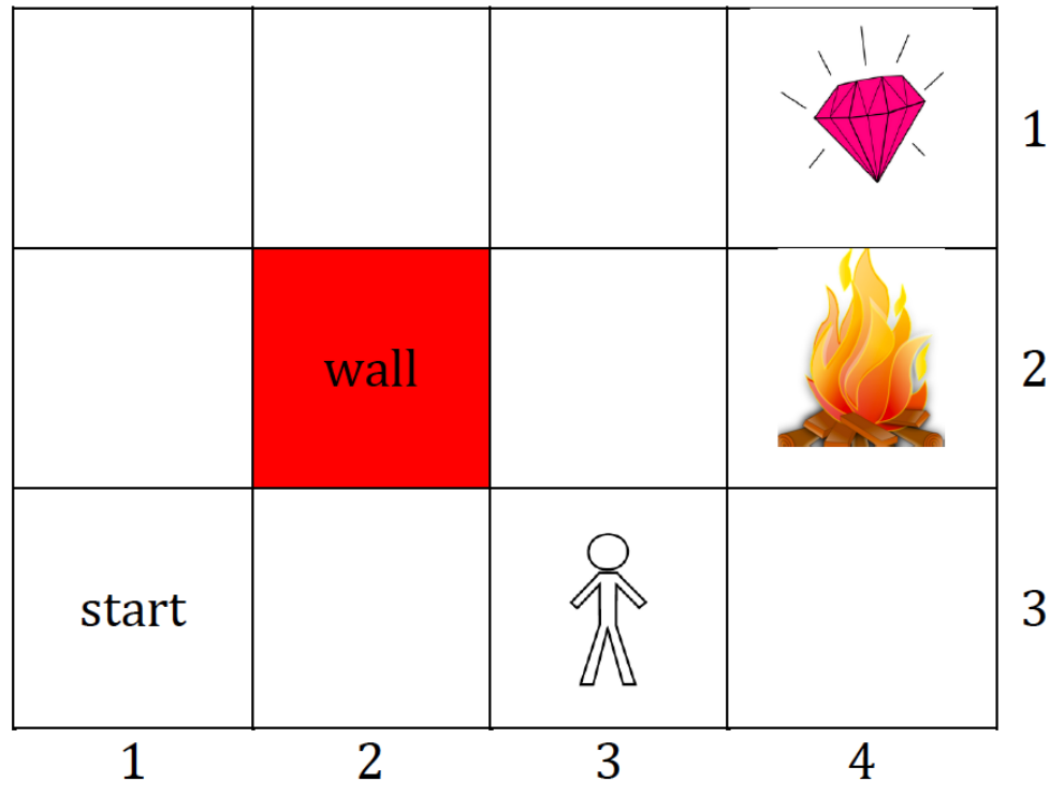
Markov Decision Process – MDP search example



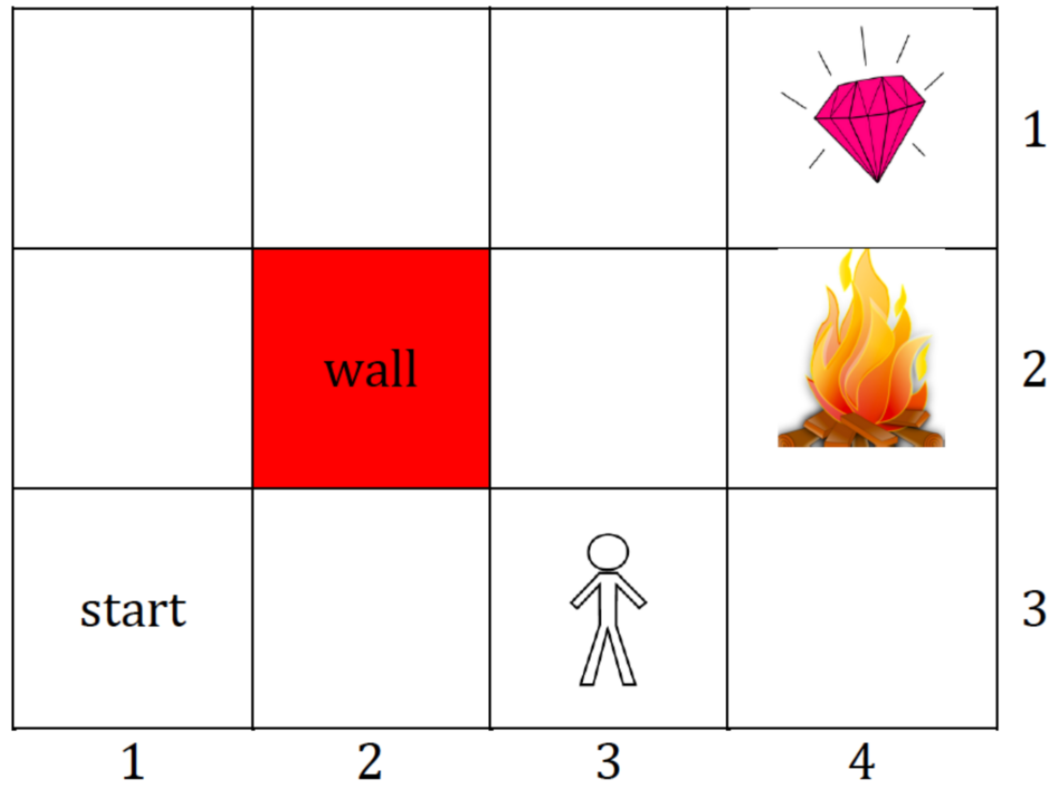
Markov Decision Process – MDP search example



Markov Decision Process – Bellman equations



Markov Decision Process – Bellman equations


wall
start
1
2
3
1
2
3
4