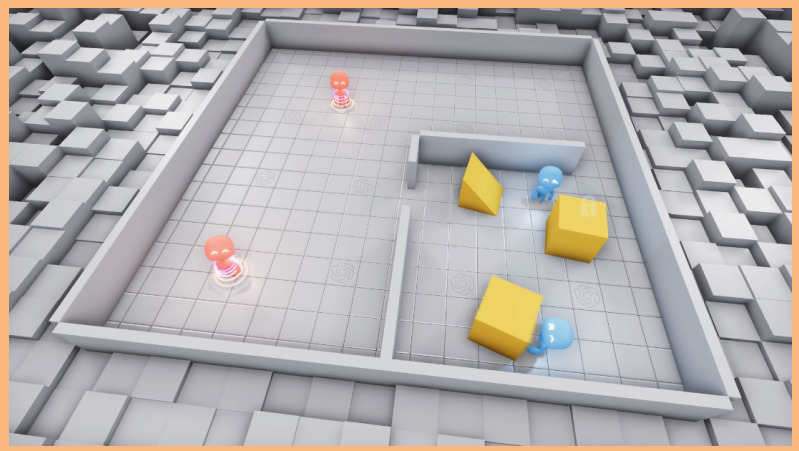
OpenAI: Hide and Seek Game

Multiple learning agents split in two teams: hiders and seekers



go to web

- Observations: other agents and blocks
- Actions: move, shift blocks
- Rewards to hiders: -1 if any hider seen by a seeker, +1 if all hiders hidden. Seekers get opposite rewards.

FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE

Observations = States

How does the example fall into our framework? The *observation* set X is *finite*. Observations in RL are also called **states**.

The initial state x_1 is distributed by $P(x_1)$ independently of y_1 (so y_1 is irrelevant). For $k > 1$, states are distributed as

$$x_k \overset{c}{\sim} P(x_k \mid x_{k-1}, y_k) \tag{1}$$

which is a special case of (??). As the distribution is the same for all $k > 1$, we will also use the index-less form

$$P(x' \mid x, y) \tag{2}$$

where x' is the state immediately following state x.

FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE

Terminal States

A state $x \in X$ is **terminal** if

$$P(x' \mid x, y) = P(x')$$

Intuitively, when a terminal state x is reached, the environment is 'restarted', i.e. the next state x' is sampled independently of x, y from the same distribution as the initial state x_1.

An **episode** is the span of interaction

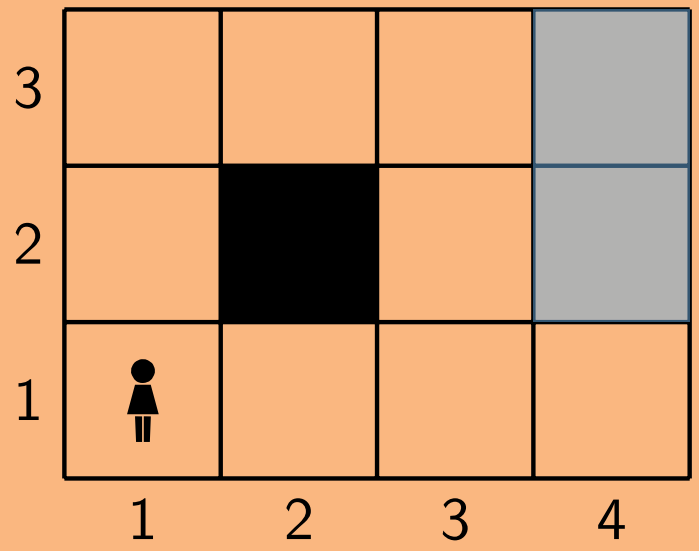
- starting with x_1 or a state immediately following a terminal state
- ending with the next terminal state.

Episodes model e.g. repeated games where terminal states may be 'game lost' or 'game won'.

FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE

Example: Grid World - States

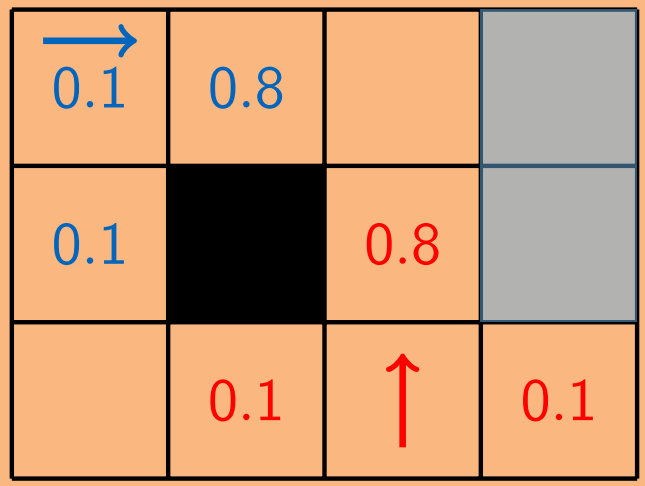
Consider a simpler example with a single agent:



X: 11 possible agent's positions on the grid.
$x_1 = (1, 1)$ with probability 1
$(4, 2), (4, 3)$ are *terminal*, always followed by state $(1, 1)$

Example: Grid World - Actions

Actions $Y = \{ \text{left}, \text{right}, \text{up}, \text{down} \}$



Uncertain effects: prescribed direction with 0.8 probability,
each of perpendicular directions with 0.1 probability

Bouncing: if new state out of grid, agent stays put

FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE

State Transitions under Policy

Due to (1), the probability of a sequence of states $x_1, x_2, \ldots$ given actions $y_1, y_2, \ldots$ is

$$P(x_1, x_2, \ldots \mid y_1, y_2, \ldots) = P(x_1)P(x_2 \mid x_1, y_2)P(x_3 \mid x_2, y_3)\ldots \quad (3)$$

Define for $k > 1$

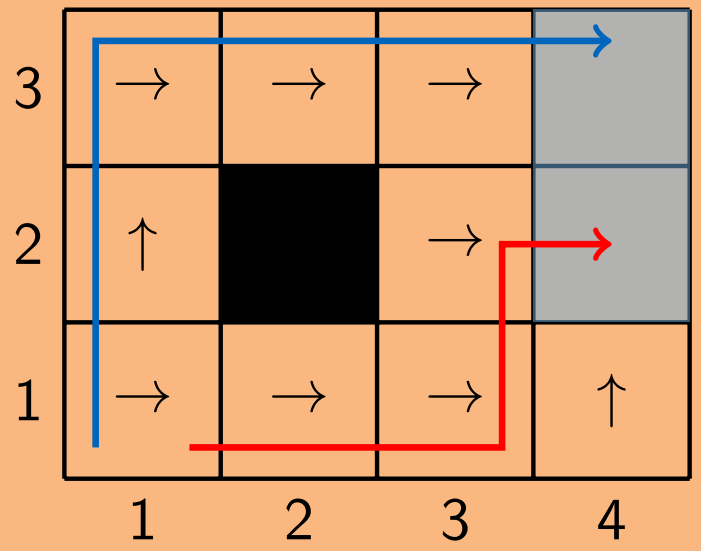
$$P^\pi(x_k \mid x_{k-1}) = P(x_k \mid x_{k-1}, \pi(x_{k-1}))$$

where π is an agent's policy. So (3) can be written as

$$P^\pi(x_1, x_2, \ldots) = P(x_1)P^\pi(x_2|x_1)P^\pi(x_3|x_2)\ldots \quad (4)$$

FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE

Example: State Transitions under Policy



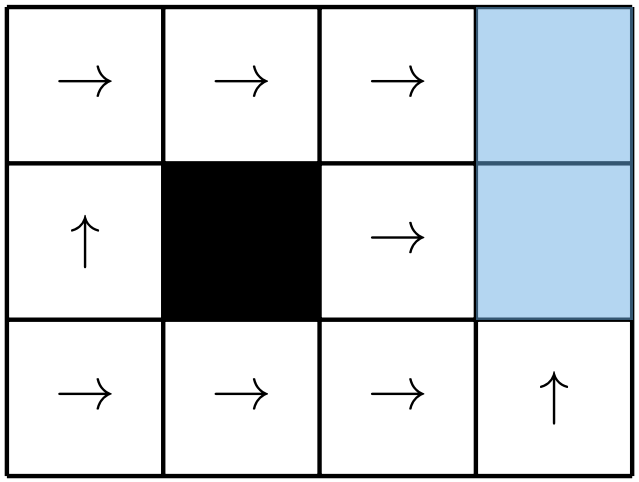
- Arrows indicate policy π for all states except terminal (irrelevant)
- Probability of blue path and red path:

$$P((1,1)) \cdot P^{\pi}((1,2) \mid (1,1)) \cdot \ldots = 1 \cdot 0.1 \cdot 0.8 \cdot 0.8 \cdot 0.8 \cdot 0.8$$
$$P((1,1)) \cdot P^{\pi}((2,1) \mid (1,1)) \cdot \ldots = 1 \cdot 0.8 \cdot 0.8 \cdot 0.1 \cdot 0.8$$

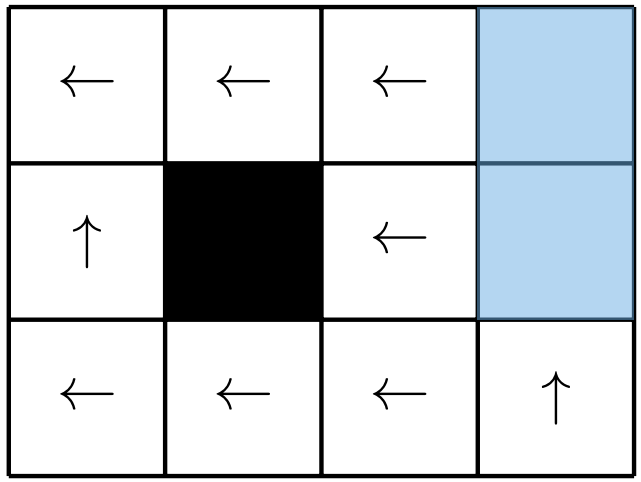
FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE

Proper Policy

A **proper policy** is a policy such that from each state $x \in X$ there is a sequence of state transitions that has non-zero probability and that ends a terminal state.



proper



not proper

Rewards

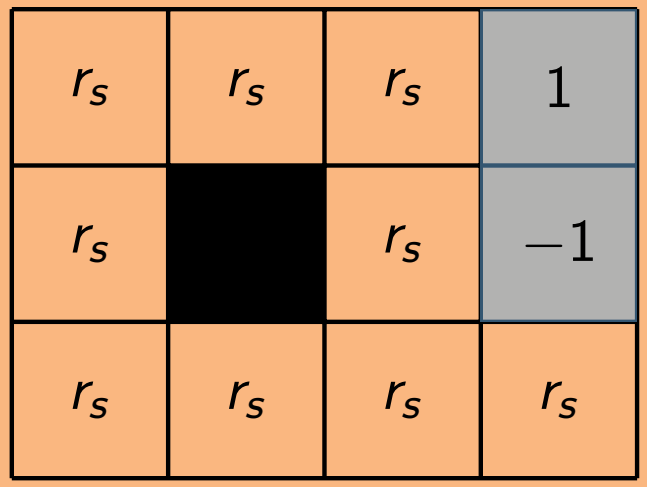
Rewards are instant, i.e., as in (??) but since x_k (1) depends on the same conditions x_{k-1}, y_k, we can assume instead a functional dependence:

$$r_k = r(x_k) \tag{5}$$

(This is because r_k can be formally made a part of x_k and 'extracted' by the function above.)

Rewards are thus a *function* of states.

Example: Grid World - Rewards



Terminal states : 'win' (1) and 'lose' (-1) rewards

Non-terminal states : a small punishment r_s (e.g., $r_s = -0.05$) for every step taken

FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE

State Utility under a Proper Policy

For a $\gamma \in [0; 1]$, define the **utility $U^\pi(x)$ of state $x \in X$ under a proper policy π** as

$$U^\pi(x) = \begin{cases} r(x) \text{ if } x \text{ is terminal} \\ r(x) + \gamma \sum_{x' \in X} P^\pi(x' \mid x) U^\pi(x') \text{ otherwise} \end{cases} \tag{6}$$

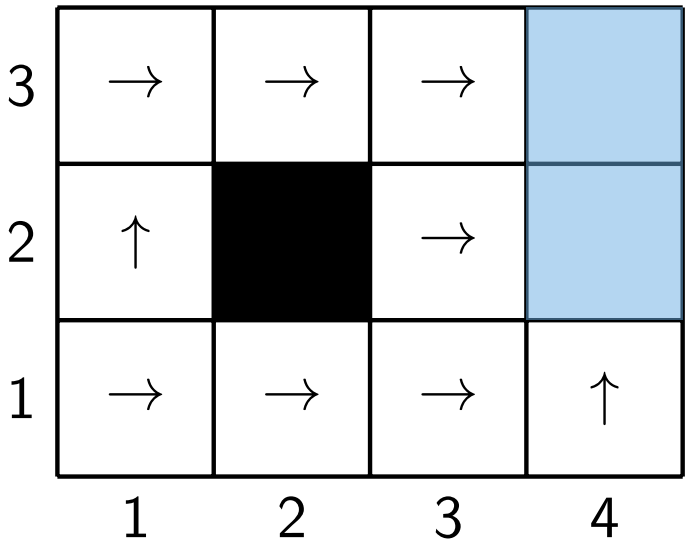
$U^\pi(x)$ is thus recursively defined as the expected sum of γ-discounted rewards in an episode starting in state x. Since an episode has a finite length, $U^\pi(x)$ converges even with $\gamma = 1$ (i.e., if γ dropped).

This is similar to U^π (??) but U^π is the expected sum of γ-discounted rewards in the entire interaction consisting of an infinite succession of episodes.

FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE

State Utility under a Proper Policy (cont'd)

Given π, $U^\pi(x)$ for all $x \in X$ can be calculated by solving $|X|$ linear equations. Example, abbreviating $U^\pi((i,j))$ as u_{ij}:



$$u_{43} = 1 \text{ (terminal)}$$

$$u_{42} = -1 \text{ (terminal)}$$

$$u_{41} = r_s + \gamma(0.8 \cdot u_{42} + 0.1 \cdot u_{31} + 0.1 \cdot u_{41})$$

etc.

Optimal Policy and State Utility

Theorem 1

For any $x, x' \in X$, $U^\pi(x)$ *and* $U^\pi(x')$ *are maximized by the same policies:*

$$\overline{\pi} = \arg\max_\pi U^\pi(x) = \arg\max_\pi U^\pi(x')$$

and if $\gamma < 1$, *these policies also maximize* U^π*:*

$$\overline{\pi} = \arg\max_\pi U^\pi$$

Intuitive proof: the best action to take in any state x *evidently does not depend on how* x *was reached - whether going from* x *or from* x'*. Thus* $\arg\max_\pi U^\pi(x) = \arg\max_\pi U^\pi(x')$*. Since* $\overline{\pi}$ *is optimal for each starting state, i.e., for each episode, the discounted sum* U^π *over all episodes is also maximized by* $\overline{\pi}$*. Hence the last equality.*

FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE

State Utility and Policy Iteration

$\overline{\pi}$ is called the **optimal policy** and the **utility of state** $x \in X$ is defined as the utility of x under the optimal policy, i.e.

$$U(x) = U^{\overline{\pi}}(x) \tag{7}$$

An optimal policy can be computed by **policy iteration**, which is similar in spirit to the EM algorithm. Start with a random initial policy π and iterate

1. $U(x) := U^{\pi}(x)$
2. compute a new policy π such that

$$\pi(x) := \arg\max_{y} \sum_{x' \in X} P(x' \mid x, y) U(x') \tag{8}$$

until the policy does not change.

FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE

Value Iteration

As an alternative to policy iteration, the following method can be used to calculate $U(x)$ and $\overline{\pi}$:

1. Calculate $U(x)$, $x \in X$ as a solution to the set of Bellman equations

$$U(x) = r(x) + \gamma \max_{y \in Y} \sum_{x' \in X} P(x' \mid x, y) U(x') \tag{9}$$

   one for each non-terminal $x \in X$ (for a terminal x, $U(x) = r(x)$). This is called **value iteration**.
2. Calculate $\overline{\pi}$ by (8).

Unlike in policy iteration, the two steps are not *iterated*. However, (9) is not linear so an iterative algorithm is needed to find its solution.

FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE

Example: Grid World - Optimal Policy and Utilities

$-.04$	$-.04$	$-.04$	1
$-.04$		$-.04$	-1
$-.04$	$-.04$	$-.04$	$-.04$

rewards

$\rightarrow$	$\rightarrow$	$\rightarrow$	
$\uparrow$		$\uparrow$	
$\uparrow$	$\leftarrow$	$\leftarrow$	$\leftarrow$

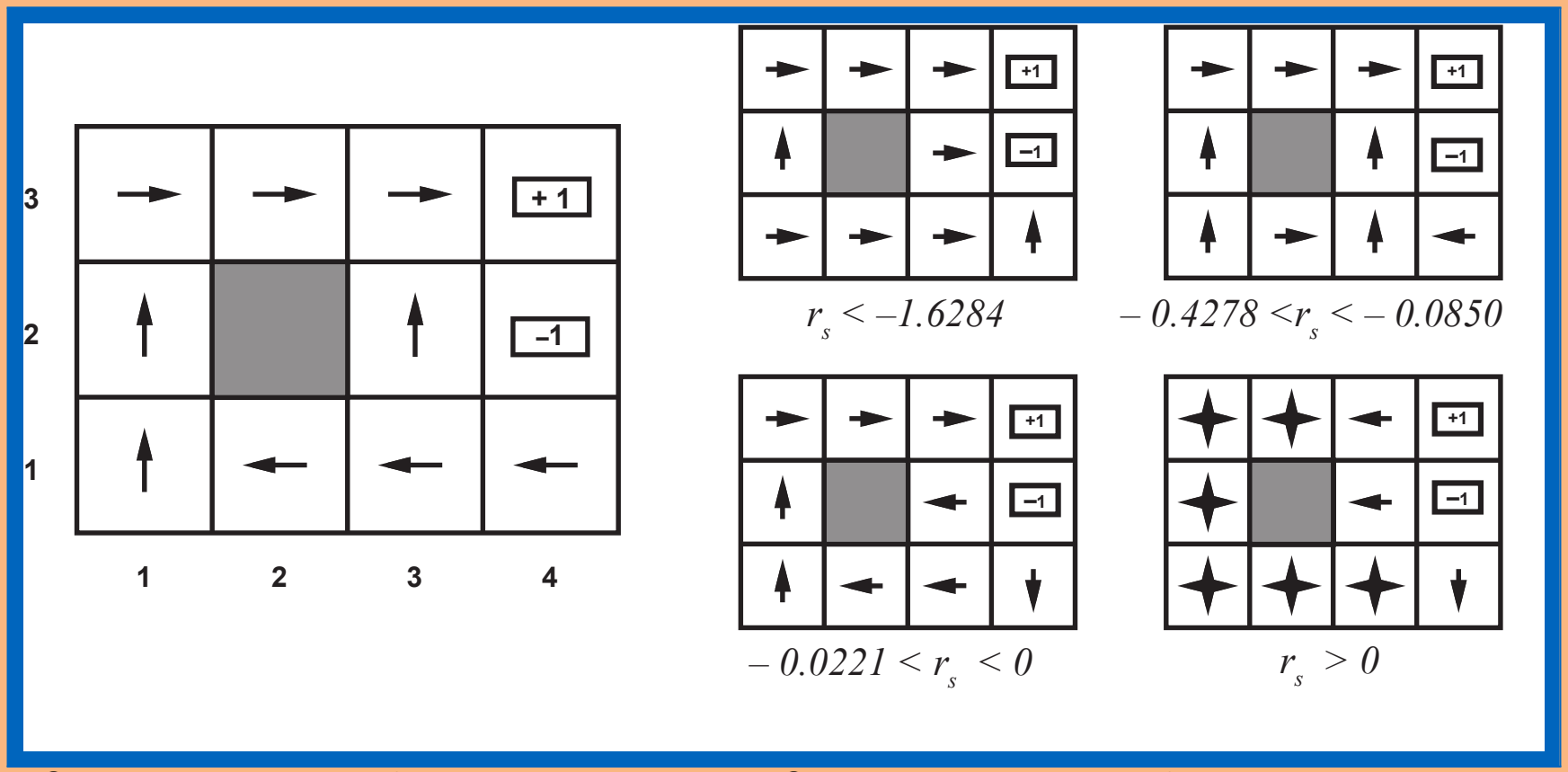
optimal policy

0.812	0.868	0.918	1
0.762		0.660	-1
0.705	0.655	0.611	0.388

state utilities

FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE

Example: Optimal Policies Under Different Rewards



Optimal policy for $\gamma = 1$ and $r_s = -0.04$ for non-terminal states.

Optimal policies for $\gamma = 1$ and different ranges of r_s for non-terminal states.

FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE

Reinforcement Learning

The goal of RL is to find a good (high-utility) policy in an *unknown* environment, i.e., the agent does not know P (2) and r (5).

After each action y following a state x, the agent observes the new state x' and receives $r(x')$.

From these (x', x, y) samples, it can learn an estimate $p(x' \mid x, y)$ of $P(x' \mid x, y)$.

- for small $|X|, |Y|$, p can be just a 3-dimensional array of relative frequencies
- larger $|X|, |Y|$ may require a parametric approach (such as a Bayes Network if independence structure available)

Learning $r(x)$ is easier - just store the reward for each state x visited.

FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE

Exploration vs. Exploitation

What actions should the agent take while collecting the (x', x, y) and r samples? Consider two options:

1. *Random* actions. Reason: a diverse sample will be collected, agent will not get stuck looping over a subset of states.
2. *Greedy* actions, i.e. optimal w.r.t. the current estimates

$$\pi(x) = \arg\max_y \sum_{x' \in X} p(x' \mid x, y) \widehat{U}(x') \tag{10}$$

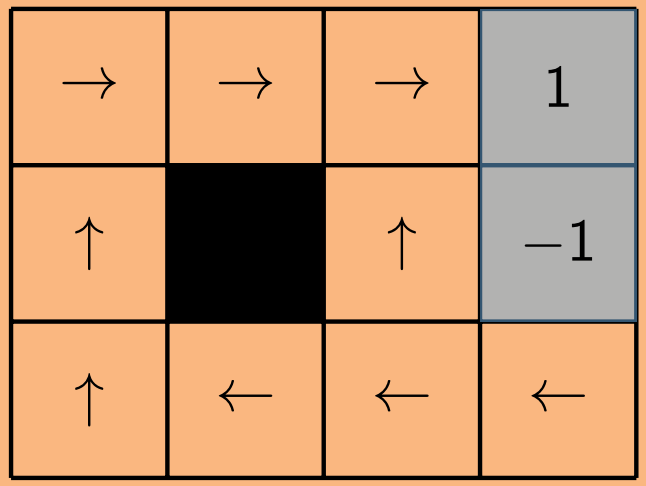
   Reason: the agent's goal is to maximize rewards.

This is the exploration-exploitation dilemma we have already investigated (although with a different estimated quantity).

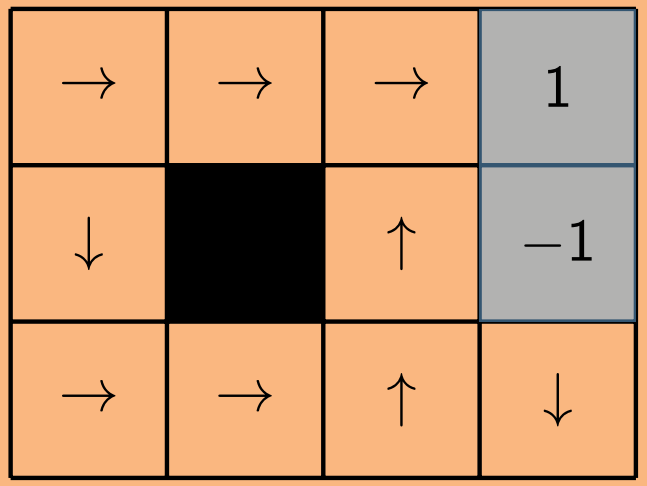
$\widehat{U}(x')$ is the estimate of $U(x')$ computed with p instead of P.

FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE

Greedy Approach



Optimal policy
($r_s = -0.04$)



Policy learned with greedy actions (10)

After each greedy action (10), the agent re-computes π by policy iteration using $p(x' \mid x, y)$ estimated from collected samples of x', x, y, r.

This process converges into a *non-optimal* policy.

FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE

GLIE Approach

To balance btw. exploration and exploitation, we can adopt this strategy, i.e. instead of (10), take a random action with probability approaching 0.

This approach is called **GLIE** = *greedy in the limit of infinite exploration*.

A popular instance of GLIE is to take action y in state x with probability

$$\frac{e^{Q(x,y)/\tau}}{\sum_{y' \in Y} e^{Q(x,y')/\tau}} \tag{11}$$

where

$$Q(x,y) = \sum_{x' \in X} p(x' \mid x, y) \widehat{U}(x') \tag{12}$$

and the *temperature* $\tau \to 0$ with $k \to \infty$.

FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE