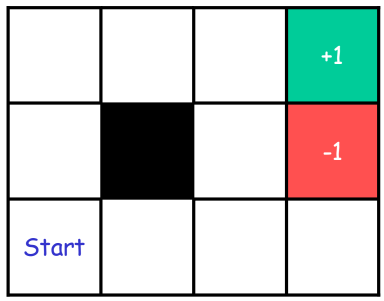
17 MAKING COMPLEX DECISIONS

- The agent's utility now depends on a sequence of decisions
- In the following 4×3 grid environment the agent makes a decision to move (U, R, D, L) at each time step
- When the agent reaches one of the goal states, it terminates
- The environment is fully observable — the agent always knows where it is



- If the environment were deterministic, a solution would be easy: the agent will always reach +1 with moves [U, U, R, R, R]
- Because actions are unreliable, a sequence of moves will not always lead to the desired outcome
- Let each action achieve the intended effect with probability 0.8 but with probability 0.1 the action moves the agent to either of the right angles to the intended direction
- If the agent bumps into a wall, it stays in the same square
- Now the sequence [U, U, R, R, R] leads to the goal state with probability $0.8^5 = 0.32768$
- In addition, the agent has a small chance of reaching the goal by accident going the other way around the obstacle with a probability $0.1^4 \times 0.8$, for a grand total of 0.32776

- A *transition model* specifies outcome probabilities for each action in each possible state
- Let $P(s' \mid s, a)$ denote the probability of reaching state s' if action a is done in state s
- The transitions are *Markovian* in the sense that the probability of reaching s' depends only on s and not the earlier states
- We still need to specify the utility function for the agent
- The decision problem is sequential, so the utility function depends on a sequence of states — an environment history — rather than on a single state
- For now, we will simply stipulate that is each state s, the agent receives a *reward* $R(s)$, which may be positive or negative

- For our particular example, the reward is -0.04 in all states except in the terminal states
- The utility of an environment history is just (for now) the sum of rewards received
- If the agent reaches the state $+1$, e.g., after ten steps, its total utility will be 0.6
- The small negative reward gives the agent an incentive to reach $[4, 3]$ quickly

- A sequential decision problem for a fully observable environment with
  - A Markovian transition model and
  - Additive rewards

  is called a *Markov decision problem* (MDP)

- An MDP is defined by the following four components:
  - Initial state s_0,
  - A set $Actions(s)$ of actions in each state,
  - Transition model $P(s' \mid s, a)$, and
  - Reward function $R(s)$

- As a solution to an MDP we cannot take a fixed action sequence, because the agent might end up in a state other than the goal
- A solution must be a policy, which specifies what the agent should do for any state that the agent might reach
- The action recommended by policy π for state s is $\pi(s)$
- If the agent has a complete policy, then no matter what the outcome of any action, the agent will always know what to do next

- Each time a given policy is executed starting from the initial state, the stochastic nature of the environment will lead to a different environment history
- The quality of a policy is therefore measured by the expected utility of the possible environment histories generated by the policy
- An optimal policy π^* yields the highest expected utility

→	→	→	+1
↑		↑	-1
↑	←	←	←

- A policy represents the agent function explicitly and is therefore a description of a simple reflex agent

- $-0.0221 < R(s) < 0$:

→	→	→	+1
↑		←	-1
↑	←	←	↓

- $-0.4278 < R(s) < -0.0850$:

→	→	→	+1
↑		↑	-1
↑	→	↑	←

- $R(s) < -1.6284$:

→	→	→	+1
↑		→	-1
↑	→	→	↑

- $R(s) > 0$:

+	+	←	+1
+		←	-1
+	+	+	↓

Utilities over time

- In case of an *infinite horizon* the agent's action time has no upper bound
- With a finite time horizon, the optimal action in a given state could change over time — the optimal policy for a finite horizon is *nonstationary*
- With no fixed time limit, on the other hand, there is no reason to behave differently in the same state at different times, and the optimal policy is stationary

- The *discounted* utility of a state sequence $s_0, s_1, s_2, \ldots$ is

$$R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \ldots,$$

where $0 < \gamma \leq 1$ is the discount factor

- When $\gamma = 1$, discounted rewards are exactly equivalent to additive rewards
- The latter rewards are a special case of the former ones
- When γ is close to 0, rewards in the future are viewed as insignificant

- If an infinite horizon environment does not contain a terminal state or if the agent never reaches one, then all environment histories will be infinitely long
- Then, utilities with additive rewards will generally be infinite
- With discounted rewards ($\gamma < 1$), the utility of even an infinite sequence is finite

- Let R_{max} be an upper bound for rewards. Using the standard formula for the sum of an infinite geometric series yields:

$$\sum_{t=0,\ldots,\infty} \gamma^t R(s_t) \leq \sum_{t=0,\ldots,\infty} \gamma^t R_{max} = R_{max}/(1 - \gamma)$$

- *Proper policy* guarantees that the agent reaches a terminal state when the environment contains such
- With proper policies infinite state sequences do not pose a problem, and we can use $\gamma = 1$ (i.e., additive rewards)

- An optimal policy using discounted rewards is

$$\pi^* = \arg\max_\pi E[\ \sum_{t=0,\ldots,\infty} \gamma^t R(s_t) \mid \pi\ ],$$

where the expectation is taken over all possible state sequences that could occur, given that the policy is executed

17.2 Value Iteration

- For calculating an optimal policy we
  - calculate the utility of each state and
  - then use the state utilities to select an optimal action in each state
- The utility of a state is the expected utility of the state sequence that might follow it
- Obviously, the state sequences depend on the policy π that is executed
- Let s_t be the state the agent is in after executing π for t steps
- Note that s_t is a random variable
- Then, executing π starting in s (= s_0) we have

$$U^\pi(s) = \mathbf{E}[\ \sum_{t=0,\ldots,\infty} \gamma^t R(s_t)]$$

- The true utility of a state $U(s)$ is just $U^{\pi^*}(s)$
- $R(s)$ is the short-term reward for being in s, whereas $U(s)$ is the long-term total reward from s onwards
- In our example grid the utilities are higher for states closer to the $+1$ exit, because fewer steps are required to reach the exit

0.812	0.868	0.912	+1
0.762		0.660	-1
0.705	0.655	0.611	0.388

The Bellman equations for utilities

- The agent may select actions using the MEU principle

$$\pi^*(s) = \arg\max_a \sum_{s'} P(s' \mid s, a)\, U(s') \qquad (*)$$

- The utility of state s is the expected sum of discounted rewards from this point onwards, hence, we can calculate it:
  - Immediate reward in state s, $R(s)$
  - \+ The expected discounted utility of the next state, assuming that the agent chooses the optimal action

$$U(s) = R(s) + \gamma \max_a \sum_{s'} P(s' \mid s, a)\, U(s')$$

- This is called the Bellman equation
- If there are n possible states, then there are n Bellman equations, one for each state

$$U(1,1) = -0.04 + \gamma \max\{\, 0.8\, U(1,2) + 0.1\, U(2,1) + 0.1\, U(1,1), \quad (U)$$
$$0.9\, U(1,1) + 0.1\, U(1,2), \quad (L)$$
$$0.9\, U(1,1) + 0.1\, U(2,1), \quad (D)$$
$$0.8\, U(2,1) + 0.1\, U(1,2) + 0.1\, U(1,1)\, \} \quad (R)$$

Using the values from the previous picture, this becomes:

$$U(1,1) = -0.04 +$$
$$\gamma \max\{\, 0.6096 + 0.0655 + 0.0705 = 0.7456, \quad (U)$$
$$0.6345 + 0.0762 = 0.7107, \quad (L)$$
$$0.6345 + 0.0655 = 0.7000, \quad (D)$$
$$0.5240 + 0.0762 + 0.0705 = 0.6707\, \} \quad (R)$$

Therefore, Up is the best action to choose

- Simultaneously solving the Bellman equations using does not work using the efficient techniques for systems of linear equations, because max is a nonlinear operation
- In the iterative approach we start with arbitrary initial values for the utilities, calculate the right-hand side of the equation and plug it into the left-hand side

  $$U_{i+1}(s) \leftarrow R(s) + \gamma \max_a \textstyle\sum_{s'} P(s' \mid s, a)\, U_i(s'),$$

  where index i refers to the utility value of iteration i
- If we apply the Bellman update infinitely often, we are guaranteed to reach an equilibrium, in which case the final utility values must be solutions to the Bellman equations
- They are also the unique solutions, and the corresponding policy is optimal