ARTIFICIAL INTELLIGENCE (E016350A)
GHENT UNIVERSITY
AY 2024/2025

Aleksandra Pizurica
Asst: Yoann Arhant
E-mail: ai@lists.ugent.be

Solutions: Markov decision processes

1. [Adapted from R&N, 3rd Ed] Sometimes MDPs are formulated with a reward function $R(s, a)$ that depends on the action taken or with a reward function $R(s, a, s')$ that also depends on the outcome state. In some formulations, rewards are assigned to the states alone: $R(s)$.

   (a) Write the Bellman equations for these formulations.

   (b) Explain compactly in words the meaning of the Bellman equation for the formulation with $R(s)$.

   (c) Show how an MDP with reward function $R(s, a, s')$ can be transformed into a different MDP with reward function $R(s, a)$, such that optimal policies in the new MDP correspond exactly to optimal policies in the original MDP.

   (d) Now do the same to convert MDPs with $R(s, a)$ into MDPs with $R(s)$.

Solution:

(a) This exercise tests the student's understanding of the formal definition of MDPs. The key here is to get the max and summation in the right place.
For $R(s, a, s')$ we have

$$U(s) = \max_a \sum_{s'} T(s, a, s') \left[R(s, a, s') + \gamma\, U(s')\right].$$

If $R(s, a, s') = R(s, a)$, then there is no dependency of the reward from the successor state s' in the sum, allowing the reward term to be moved outside the sum. The corresponding expression is

$$U(s) = \max_a [R(s, a) + \gamma \sum_{s'} T(s, a, s') U(s')]$$

where we use that for any given s and a, $\sum_{s'} T(s, a, s') = 1$.

Solution:

(a) (Continued) Finally, if $R(s, a, s') = R(s)$, then the immediate reward does not depend on the action in $\max_a$ either, and can be moved outside the maximization too. Then we easily obtain the following expression

$$U(s) = R(s) + \gamma \max_a \sum_{s'} T(s, a, s')U(s'),$$

where we use the algebraic property $\max_a(g + f(a)) = g + \max_a f(a)$, which holds whenever g is independent of a.

(b) The Bellman equation in the formulation with $R(s)$ reads directly as follows: *The utility of a state is the immediate reward for that state plus the discounted expected utility of the next state, assuming that the agent chooses the optimal action.*

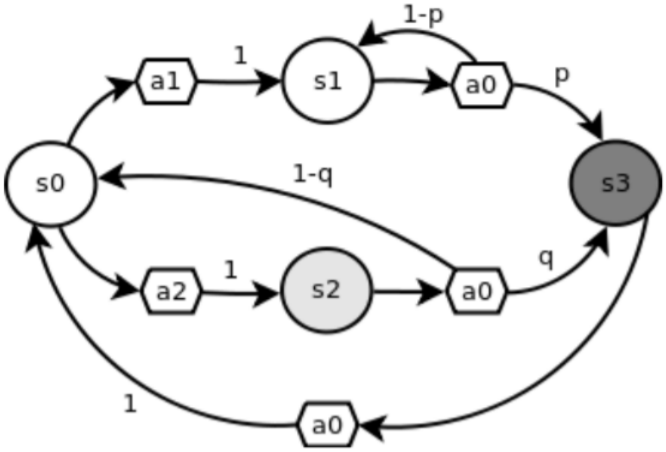
(c) There are a variety of solutions here. One is to create a "pre-state" $pre(s, a, s')$ for every s, a, s', such that executing a in s leads not to s' but to $pre(s, a, s')$. In this state is encoded the fact that the agent came from s and did a to get here. From the pre-state, there is just one action b that always leads to s'. Let the new MDP have transition T', reward R', and discount γ'. Then

$$\begin{aligned}
&T'(s, a, pre(s, a, s')) = T(s, a, s')\\
&T'(pre(s, a, s'), b, s') = 1\\
&R'(s, a) = 0\\
&R'(pre(s, a, s'), b) = \gamma^{-\frac{1}{2}} R(s, a, s')\\
&\gamma' = \gamma^{\frac{1}{2}}
\end{aligned}$$

(d) Following the idea of part (c), we can create a "post-state" $post(s, a)$ for every s, a, such that

$$\begin{aligned}
&T'(s, a, post(s, a)) = 1\\
&T'(post(s, a), b, s') = T(s, a, s')\\
&R'(s) = 0\\
&R'(post(s, a)) = \gamma^{-\frac{1}{2}} R(s, a)\\
&\gamma' = \gamma^{\frac{1}{2}}
\end{aligned}$$

2. Consider the infinite-horizon Markov decision process $\mathcal{M}$ represented by figure below with discount factor $\gamma \in [0, 1)$. States are represented by circles and actions by hexagons.



The numbers on the arrows from actions to states represent the transition probability, for instance, $T(s_2, a_0, s_3) = P(s_3 \mid s_2, a_0) = q$. Each of the parameters p and q are in the interval $[0, 1]$. The reward is 10 for state s_3, 1 for state s_2 and 0 otherwise.

(a) List all the possible policies for $\mathcal{M}$.

(b) Show the equation representing the optimal value function for each state of $\mathcal{M}$, i.e., $U^*(s_0)$, $U^*(s_1)$, $U^*(s_2)$ and $U^*(s_3)$.

(c) Is there a value for p such that for all $\gamma \in [0, 1)$ and $q \in [0, 1]$, $\pi^*(s_0) = a_2$? Explain.

(d) Is there a value for q such that for all $\gamma \in [0, 1)$ and $p > 0$, $\pi^*(s_0) = a_1$? Explain.

(e) Using $p = q = 0.25$ and $\gamma = 0.9$, compute π^* and U^* for all states of $\mathcal{M}$.

Solution:

(a) There are two possible policies:

	s_0	s_1	s_2	s_3
π_1	a_1	a_0	a_0	a_0
π_2	a_2	a_0	a_0	a_0

(b) Since the reward depends only on the state and not on the transition, from Exercise 1 we have the following:

$$\begin{aligned}
U^*(s_0) &= 0 + \gamma \max_{a \in \{a_1, a_2\}} \sum_{s'} T(s_0, a, s')U^*(s') = \gamma \max\{U^*(s_1), U^*(s_2)\} \\
U^*(s_1) &= 0 + \gamma \max_{a \in \{a_0\}} \sum_{s'} T(s_1, a, s')U^*(s') = \gamma \ [(1-p)U^*(s_1) + pU^*(s_3)] \\
U^*(s_2) &= 1 + \gamma \max_{a \in \{a_0\}} \sum_{s'} T(s_2, a, s')U^*(s') = 1 + \gamma \left[(1-q)U^*(s_0) + qU^*(s_3)\right]
\end{aligned}$$

Solution:

(b) (Continued)

$$U^*(s_3) \;=\; 10 + \gamma \max_{a \in \{a_0\}} \sum_{s'} T(s_3, a, s') U^*(s') = 10 + \gamma U^*(s_0)$$

(c) Note that

$$\pi^*(s_0) = \gamma \operatorname*{arg\,max}_{a \in \{a_1, a_2\}} \sum_{s'} T(s, a, s') U^*(s'),$$

therefore if $\gamma = 0$ then a_1 and a_2 are tied and π^* is not unique, so we cannot guarantee that $\pi^*(s_0) = a_2$. For $\gamma > 0$, $\pi^*(s_0) = a_2$ if and only if $U^*(s_2) > U^*(s_1)$ for all $q \in [0, 1]$. If $p = 0$, then $U^*(s_1) = \gamma U^*(s_1) = 0$ and since

$$U^*(s_2) = 1 + \gamma \left[(1 - q) U^*(s_0) + q U^*(s_3)\right] \geq 1,$$

we have that $U^*(s_2) > U^*(s_1)$ and $\pi^*(s_0) = a_2$ for all $\gamma \in (0, 1)$ and $q \in [0, 1]$.

(d) No. Since

$$U^*(s_2) = 1 + \gamma \left[(1 - q) U^*(s_0) + q U^*(s_3)\right] \geq 1,$$

then $\pi^*(s_0) = a_1$ if and only if $U^*(s_1) > U^*(s_2) \geq 1$ for all $\gamma \in [0, 1)$ and $p > 0$. We have that

$$U^*(s_1) = \gamma \left[(1 - p) U^*(s_1) + p U^*(s_3)\right] = \frac{\gamma \, p \, U^*(s_3)}{1 - \gamma(1 - p)}.$$

Therefore we can always find a value for γ sufficiently small such that $U^*(s_1) < 1$. Notice that if γ equals 0, then π^* is not unique (as in the previous item).

(e) We obtain the following:

	s_0	s_1	s_2	s_3
U^*	14.1846	15.7608	15.6969	22.7661
π^*	a_1	a_0	a_0	a_0

We can either solve the recursion we obtained in (b) or implement value iteration. In the latter case, we also have to specify the error between U^t and U^*.

It is simpler to solve the recursion in (b). Directly we have that

$$\begin{aligned} U^*(s_1) &= 6.92 + 0.623 \, U^*(s_0) \\ U^*(s_2) &= 3.25 + 0.877 \, U^*(s_0) \\ U^*(s_3) &= 10 + 0.9 \, U^*(s_0) \end{aligned}$$

from where follows that

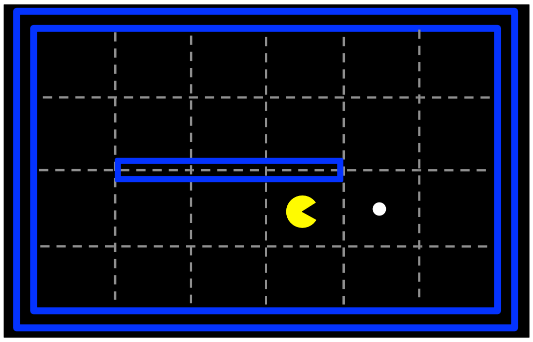
$$U^*(s_0) = \max\{6.92 + 0.623 \, U^*(s_0), 3.25 + 0.877 \, U^*(s_0)\}$$

and thus $U^*(s_0) = 14.1846$ and the action which lead to this value corresponds with $a = a_1$. Now we easily obtain $U^*(s_1)$, $U^*(s_2)$ and $U^*(s_3)$ with the only available action $a = a_0$.

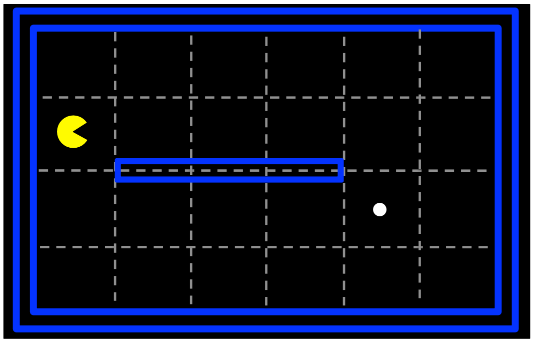
3. Let us consider a Pacman agent in a deterministic environment. A goal state is reached when there are no remaining food pellets on the board. Pacman's available actions are $\{N, S, E, W\}$, but Pacman **can not** move into a wall. Whenever Pacman eats a food pellet he receives a reward of $+1$.

   Assume that pacman eats a food pellet as soon as he occupies the location of the food pellet - i.e., the reward is received for the transition into the square with the food pellet.

   Consider the particular Pacman board states shown below. Throughout this problem assume that $U_0(s) = 0$ for all states s. Let the discount factor be $\gamma = 1$.



State A



State B

   (a) What is the optimal value of state A, $U^*(A)$?

   (b) What is the optimal value of state B, $U^*(B)$?

   (c) At what iteration, k, will $U_k(B)$ first be non-zero?

   (d) How do the optimal q-state values of moving W and E from state A compare? (*choose one*)

   ○ $Q^*(A, W) > Q^*(A, E)$ ○ $Q^*(A, W) < Q^*(A, E)$ ○ $Q^*(A, W) = Q^*(A, E)$

   (e) If we use this MDP formulation, is the policy found guaranteed to produce the shortest path from pacman's starting position to the food pellet? If not, how could you modify the MDP formulation to guarantee that the optimal policy found will produce the shortest path from pacman's starting position to the food pellet?

Solution:

(a) $U^*(A) = 1$.

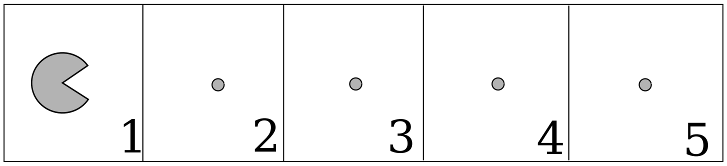
(b) $U^*(B) = 0 + 1 \cdot 0 + 1 \cdot 0 + 0 \cdot 1 + 1 \cdot 1 = 1$.
The reason the answers are the same for both (b) and (a) is that there is no penalty for existing. With a discount factor of $\gamma = 1$, eating the food at any future step is just as valuable as eating it on the next step. An optimal policy will definitely find the food, so the optimal value of any state is always 1.
Note: If a discount factor $\gamma \neq 1$ then the answers wouldn't be the same (see (e)).

(c) 5
The value function at iteration k is equivalent to the maximum reward possible within k steps of the state in question, B. Since the food pellet is exactly 5 steps away from Pacman in state B, $U_5(B) = 1$ and $U_{K<5}(B) = 0$.
Note that you only receive your reward the moment you enter the box. So for $k = 1$ the boxes around the dot receive value 1. In iteration $k = 0$ everything is set to 0. In iteration $k = 1$ each state gets the value of the reward of its best action plus the utility of the state that follows from that action as specified in the assignment. The neighbors of the dot will first obtain the reward 1 according to the value iteration. And thus we have indeed that $k = 5$.
Compare it with the Grid world example from the theory lesson on MDP.
Note: The time-limited value for a state s with a time-limit of k time steps is denoted by $U_k(s)$, and represents the maximum expected utility attainable from s given that the Markov decision process under consideration terminates in k time steps. For every $s \in S$ we initialize $U_0(s) = 0$.

(d) $Q^*(A, W) = Q^*(A, E)$
Once again, since $\gamma = 1$, the optimal value of every state is the same, since the optimal policy will eventually eat the food.

(e) No. The Q-values for going $West$ and $East$ from state A are equal so there is no preference given to the shortest path to the goal state. Adding a negative living reward (example: -1 for every time step) will help differentiate between two paths of different lengths. Setting $\gamma < 1$ will make rewards seen in the future worth less than those seen right now, incentivizing Pacman to arrive at the goal as early as possible.

4. Pacman is in a bonus level! With no ghosts around, he can eat as many dots as he wants. He is in the 5×1 grid shown. The cells are numbered from left to right as $1, \ldots, 5$. In cells 1 through 4, the actions available to him are to move *Right* (R) or to *Fly* (F) out of the bonus level. The action *Right* deterministically lands Pacman in the cell to the right (and he eats the dot there), while the *Fly* action deterministically lands him in a terminal state and ends the game. From cell 5, *Fly* is the only action. Eating a dot gives a reward of 10, while flying out gives a reward of 20. Pacman starts in the leftmost cell (cell 1).



We write this as an MDP where the state is the cell that Pacman is in. The discount is γ.

Consider the following 3 policies:

$$\begin{aligned} \pi_0(s) &= F \text{ for all } s \\ \pi_1(s) &= R \text{ if } s \leq 3,\ F \text{ otherwise} \\ \pi_2(s) &= R \text{ if } s \leq 4,\ F \text{ otherwise} \end{aligned}$$

(a) Assume $\gamma = 1.0$. Calculate:

1. $U^{\pi_0}(1) =$ 2. $U^{\pi_1}(1) =$ 3. $U^{\pi_2}(1) =$ 4. $U^*(1) =$

(b) Now consider an arbitrary value for γ.

1. Does there exist a value for γ such that π_0 is strictly better than both π_1 and π_2? If yes, give a value for γ. If no, write *none*.

2. Does there exist a value for γ such that π_1 is strictly better than both π_0 and π_2? If yes, give a value for γ. If no, write *none*.

3. Does there exist a value for γ such that π_2 is strictly better than both π_0 and π_1? If yes, give a value for γ. If no, write *none*.

Solution:

(a) $\gamma = 1.0$ so we obtain:

1. From here the pacman has to fly out so the reward is 20 and then $U^{\pi_0}(1) = 20$.
2. Here the pacman moves right three times and then flies out, so we obtain that $U^{\pi_1}(1) = 10 + 10 + 10 + 20 = 50$.
3. Similarly, $U^{\pi_2}(1) = 60$.
4. The optimal value of state 1 is $U^*(1) = 60$.

(b) 1. Yes. $0 \leq \gamma < \frac{1}{2}$. Let us explain how to get this answer. Assuming we start at state 1:

$$\begin{array}{rcl} U^{\pi_0}(1) & > & U^{\pi_2}(1) \\ 20 & > & 10 + 10 \cdot \gamma + 10 \cdot \gamma^2 + 10 \cdot \gamma^3 + 20 \cdot \gamma^4 \\ 0 & > & -1 + \gamma + \gamma^2 + \gamma^3 + 2\gamma^4 \end{array}$$

We have to study the function $f(\gamma) = -1 + \gamma + \gamma^2 + \gamma^3 + 2\gamma^4$ over the interval $[0, 1]$, with its derivative $f'(\gamma) = 1 + 2\gamma + 3\gamma^2 + 8\gamma^3$ we can get the following variation table :

γ	$0 \qquad\qquad 1$
$f'(\gamma)$	$+$
$f(\gamma)$	↗

Then looking at the bounds of $f(\gamma)$ on the interval $[0, 1]$;

$$\begin{array}{rcl} f(0) & = & -1 \\ f(1) & = & 4 \end{array}$$

and as $f \in C^0([0, 1])$ and f strictly monotonous on $[0, 1]$ as a consequence of the bijection theorem, we get that $\exists! x_0 \in [0, 1] / f(x_0) = 0$.

We know there is one unique solution but we need an intuition to find its value. From Zeno's paradoxes geometric serie, we know :

$$\begin{array}{rcl} \sum_{n=1}^{\infty} \frac{1}{2^n} & = & \frac{1}{1 - \frac{1}{2}} - 1 \\ & = & 1 \end{array}$$

Solution:

(b)

1. (Continued) Hence stopping the serie at any order of $n = k$ will yield the residual :

$$\sum_{n=k}^{\infty} \gamma^n \quad = \quad \frac{\gamma^{k+1}}{1-\gamma}$$

with $\gamma = \frac{1}{2}$,

$$\sum_{n=k}^{\infty} \frac{1}{2^n} \quad = \quad \frac{1}{2^k}$$

and then we have $f\left(\frac{1}{2}\right) = 0$:

γ	0	$\frac{1}{2}$	1
$f'(\gamma)$		$+$	
$f(\gamma)$	-1	↗ 0 ↗	4

Finally,

$$\begin{aligned} U^{\pi_0}(1) > U^{\pi_2}(1) \quad &\iff \quad f(\gamma) < 0 \\ &\iff \quad 0 \leq \gamma < \frac{1}{2} \end{aligned}$$

and similarly for $U^{\pi_0}(1) > U^{\pi_1}(1)$ we get $1 > 2\gamma^3 + \gamma^2 + \gamma$, which in both cases gives us that $\gamma < \frac{1}{2}$.

2. None! Similarly, like in 1. from $U^{\pi_1}(1) > U^{\pi_0}(1)$ we get that $\gamma > \frac{1}{2}$. From $U^{\pi_1}(1) > U^{\pi_2}(1)$ follows that $\gamma < \frac{1}{2}$. Then we obtain there doesn't exists γ such that the both conditions are satisfied.

3. Yes. $\frac{1}{2} < \gamma \leq 1$. From before, we know that to beat π_0, we must have $\gamma > \frac{1}{2}$. Writing out $U^{\pi_1}(1) < U^{\pi_2}(1)$ will give the same inequality.

5. It is useful to know that partially observable MDP (POMDP) can be solved by defining an *observable* MDP on *belief states*. Furthermore, it can be shown that an optimal policy for this MDP $\pi^*(b)$ is also an optimal policy for the original MDP. In other words, solving a POMDP on a physical state space can be reduced to solving an MDP on the corresponding belief-state space. To specify the needed MDP on the belief space, we need to define the corresponding transition model and a reward function.

   (a) How can we calculate the transition model for the belief states $P(b' \mid b, a)$ from the available information, which includes the transition model on the physical states and the the sensor model? Write the concrete expression.

   (b) Write the expression for the expected reward if the agent does a in belief state b.

Solution:

(a) The desired transition model is the probability of reaching b' from b, given action a, which we write as $P(b' \mid b, a)$. This probability needs to be calculated taking into account all possible evidence values e as follows:

$$P(b' \mid b, a) = P(b' \mid a, b) = \sum_e P(b' \mid e, a, b) P(e \mid a, b).$$

The probability of perceiving e, given that a was performed starting in belief state b, is given by summing over all the actual states s' that the agent might reach:

$$\begin{aligned} P(e \mid a, b) &= \sum_{s'} P(e \mid a, s', b) P(s' \mid a, b) \\ &= \sum_{s'} P(e \mid s') P(s' \mid a, b) \\ &= \sum_{s'} P(e \mid s') \sum_s P(s' \mid s, a) b(s). \end{aligned}$$

Finally, we obtain that

$$P(b' \mid b, a) = \sum_e P(b' \mid e, a, b) \sum_{s'} P(e \mid s') \sum_s P(s' \mid s, a) b(s)$$

where $P(b' \mid e, a, b)$ is 1 if $b' = \text{FORWARD}(b, a, e)$ and 0 otherwise.

(b) We can also define a reward function for belief states, more precisely, the expected reward if the agent does a in belief state b is

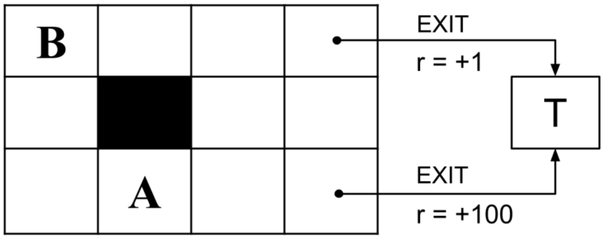
$$\rho(b, a) = \sum_s b(s) \sum_{s'} P(s' \mid s, a) R(s, a, s').$$

Together, $P(b' \mid b, a)$ and $\rho(b, a)$ define an observable MDP on the space of belief states.

6. Let us consider the following problem which takes place in various instances of a grid world MDP. Shaded cells represent walls. In all states, the agent has available actions $\{\uparrow, \downarrow, \rightarrow, \leftarrow\}$. Performing an action that would transition to an invalid state (outside the grid or into a wall) results in the agent remaining in its original state. In states with an arrow coming out, the agent has an additional action $EXIT$. In the event that the $EXIT$ action is taken, the agent receives the labeled reward and ends the game in the terminal state T. Unless otherwise stated, all other transitions receive no reward, and all transitions are deterministic.

   For all parts of the problem, assume that value iteration begins with all states initialized to zero, i.e., $U_0(s) = 0$, for every s. Let the discount factor be $\gamma = \frac{1}{2}$.

   (a) Suppose that we are performing value iteration on the grid world MDP below.



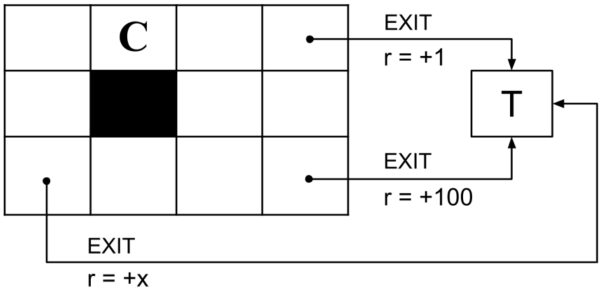
   (i) Find in the optimal values for A and B, i.e., determine $U^*(A)$ and $U^*(B)$.

   (ii) After how many iterations k will we have $U_k(s) = U^*(s)$ for all states s? If it never occurs, write "never". Explain your reasoning.

   (iii) Suppose that we wanted to re-design the original reward function $R(s, a, s')$. For which of the following new reward functions would the optimal policy remain unchanged?

   ○ $R_1(s, a, s') = 10\,R(s, a, s')$
   ○ $R_2(s, a, s') = 1 + R(s, a, s')$
   ○ $R_3(s, a, s') = (R(s, a, s'))^2$
   ○ $R_4(s, a, s') = -1$
   ○ None

   (b) For the following problem, we add a new state in which we can take the $EXIT$ action with a reward of $+x$.



   (i) For what values of x is it guaranteed that our optimal policy π^* has $\pi^*(C) = \leftarrow$? Write $+\infty$ and $-\infty$ if there is no upper or lower bound, respectively. Write the upper and lower bounds on the following lines:

$$\underline{\hspace{3cm}} < x < \underline{\hspace{3cm}}$$

(ii) For what values of x does value iteration take the minimum number of iterations k to converge to U^* for all states? Write $+\infty$ and $-\infty$ if there is no upper or lower bound, respectively.Write the upper and lower bounds on the following lines:

$$\_\_\_\_\_\_\_\_\_\_\_\_ \leq x \leq \_\_\_\_\_\_\_\_\_\_\_\_$$

(iii) What is the minimum number of iterations k until U_k has converged to U^* for all states.

Solution:

(a) (i) We know that we can only claim reward from going through one of the two exists since all other transitions receive no reward and we know that the discount factor is $\gamma = 0.5$. Moreover, it is clear that the optimal policy will not go through the top exit. So without a living reward and if the only way to gain a reward is through the bottom exit, we can easily obtain that $U^*(A) = 25$ and $U^*(B) = \frac{25}{8}$, since the optimal values will decrease by a factor $\frac{1}{2}$ as we go further away from the bottom exit. Indeed, $U^*(A) = \frac{1}{2^2} \cdot 100 = 25$ and $U^*(B) = \frac{1}{2^5} \cdot 100 = \frac{25}{8}$.

(ii) Here we are asked how many steps will it take us to calculate the values $U_k(s)$, for every state s. Here it is enough to calculate the number of steps only for the state B (since it is the furthest one from the bottom exit). $U_k(B)$ is essentially the number of steps needed to get from B to the bottom exit. So we obtain that $k = 6$.

(iii) R_1: Scaling the reward function does not affect the optimal policy, as it scales all Q-values by 10, which retains ordering.
R_2 : Since reward is discounted, the agent would get more reward exiting then infinitely cycling between states.
R_3 : The only positive reward remains to be from exiting state +100 and +1, so the optimal policy doesn't change.
R_4: With negative reward at every step, the agent would want to exit as soon as possible, which means the agent would not always exit at the bottom-right square.

(b) (i) We want the optimal policy at state C to be to go left and we will find a bound for x for which that is guaranteed to be true. It essentially asks us, which path will grant us the better reward. In other words, we want to choose now the new exit in the bottom left corner over the one in the bottom right corner. We obtain that the value of going left is $Q(C, \leftarrow) = \frac{1}{2^3} \cdot x = \frac{x}{8}$ and similarly that $Q(C, \rightarrow) = \frac{1}{2^4} \cdot 100 = \frac{25}{4}$. So the question is for which value of x is $\frac{x}{8} > \frac{25}{4}$ which implies that

$$\underline{\qquad 50 \qquad} < x < \underline{\qquad +\infty \qquad}$$

(ii) The two states that will take the longest for value iteration to become nonzero from either $+x$ or +100, are states C, and the state to the right of C, let us denote it by D. C will become nonzero at iteration 4 from $+x$, and D will become nonzero at iteration 4 from +100. We must bound x so that the optimal policy at D does not choose to go to $+x$, or else value iteration will take 5 iterations. Similar reasoning for C and +100. Then our inequalities are

$$\frac{1}{2^3} \cdot x \geq \frac{1}{2^4} \cdot 100 \qquad \text{and} \qquad \frac{1}{2^4} \cdot x \leq \frac{1}{2^3} \cdot 100.$$

Simplifying, we get the following bound on x:

$$\underline{\qquad 50 \qquad} \leq x \leq \underline{\qquad 200 \qquad}$$

(iii) As already explained in (a)-(ii) and (b)-(ii) we obtain that $k = 4$.