

### Q1. MDP

Pacman is using MDPs to maximize his expected utility. In each environment:

- Pacman has the standard actions {North, East, South, West} unless blocked by an outer wall
  - There is a reward of 1 point when eating the dot (for example, in the grid below,  $R(C, South, F) = 1$ )
  - The game ends when the dot is eaten
- (a) Consider a the following grid where there is a single food pellet in the bottom right corner ( $F$ ). The **discount** factor is 0.5. There is no living reward. The states are simply the grid locations.

A	B	C
D	E	F ○

(i) What is the optimal policy for each state?

State	$\pi(state)$
A	East or South
B	East or South
C	South
D	East
E	East

(ii) What is the optimal value for the state of being in the upper left corner ( $A$ )? Reminder: the discount factor is 0.5.

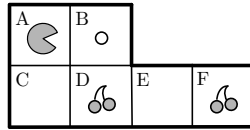
$V^*(A) = 0.25$

k	V(A)	V(B)	V(C)	V(D)	V(E)	V(F)
0	0	0	0	0	0	0
1	0	0	1	0	1	0
2	0	0.5	1	0.5	1	0
3	0.25	0.5	1	0.5	1	0
4	0.25	0.5	1	0.5	1	0

(iii) Using value iteration with the value of all states equal to zero at  $k=0$ , for which iteration  $k$  will  $V_k(A) = V^*(A)$ ?

$k = 3$  (see above)

- (b) Consider a new Pacman level that begins with cherries in locations  $D$  and  $F$ . Landing on a grid position with cherries is worth 5 points and then the cherries at that position disappear. There is still one dot, worth 1 point. The game still only ends when the dot is eaten.



- (i) With no discount ( $\gamma = 1$ ) and a living reward of -1, what is the optimal policy for the states in this level's state space?

State	$\pi(state)$
A, $D_{\text{Cherry}}=\text{true}$ , $F_{\text{Cherry}}=\text{true}$	South
A, $D_{\text{Cherry}}=\text{true}$ , $F_{\text{Cherry}}=\text{false}$	South
A, $D_{\text{Cherry}}=\text{false}$ , $F_{\text{Cherry}}=\text{true}$	East
A, $D_{\text{Cherry}}=\text{false}$ , $F_{\text{Cherry}}=\text{false}$	East
C, $D_{\text{Cherry}}=\text{true}$ , $F_{\text{Cherry}}=\text{true}$	East
C, $D_{\text{Cherry}}=\text{true}$ , $F_{\text{Cherry}}=\text{false}$	East
C, $D_{\text{Cherry}}=\text{false}$ , $F_{\text{Cherry}}=\text{true}$	East
C, $D_{\text{Cherry}}=\text{false}$ , $F_{\text{Cherry}}=\text{false}$	North/East
D, $D_{\text{Cherry}}=\text{false}$ , $F_{\text{Cherry}}=\text{true}$	East
D, $D_{\text{Cherry}}=\text{false}$ , $F_{\text{Cherry}}=\text{false}$	North
E, $D_{\text{Cherry}}=\text{true}$ , $F_{\text{Cherry}}=\text{true}$	East
E, $D_{\text{Cherry}}=\text{true}$ , $F_{\text{Cherry}}=\text{false}$	West
E, $D_{\text{Cherry}}=\text{false}$ , $F_{\text{Cherry}}=\text{true}$	East
E, $D_{\text{Cherry}}=\text{false}$ , $F_{\text{Cherry}}=\text{false}$	West
F, $D_{\text{Cherry}}=\text{true}$ , $F_{\text{Cherry}}=\text{false}$	West
F, $D_{\text{Cherry}}=\text{false}$ , $F_{\text{Cherry}}=\text{false}$	West

- (ii) With no discount ( $\gamma = 1$ ), what is the range of living reward values such that Pacman eats exactly one cherry when starting at position  $A$ ?

Valid range for the living reward is  $(-2.5, -1.25)$ .

Let  $x$  equal the living reward.

The reward for eating zero cherries  $\{A, B\}$  is  $x + 1$  (one step plus food).

The reward for eating exactly one cherry  $\{A, C, D, B\}$  is  $3x + 6$  (three steps plus cherry plus food).

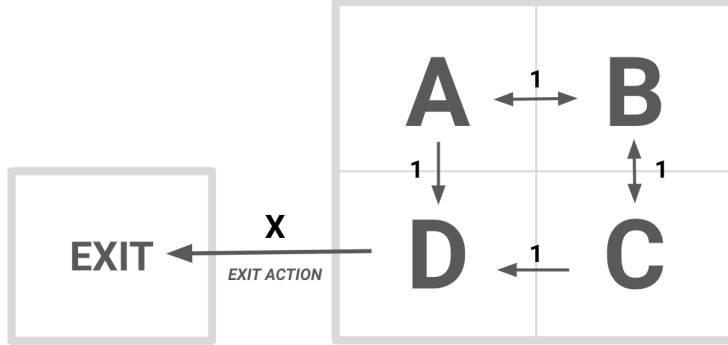
The reward for eating two cherries  $\{A, C, D, E, F, E, D, B\}$  is  $7x + 11$  (seven steps plus two cherries plus food).

$x$  must be greater than -2.5 to make eating at least one cherry worth it ( $3x + 6 > x + 1$ ).

$x$  must be less than -1.25 to eat less than one cherry ( $3x + 6 > 7x + 11$ ).

## Q2. Strange MDPs

In this MDP, the available actions at **state A, B, C** are *LEFT*, *RIGHT*, *UP*, and *DOWN* unless there is a wall in that direction. The only action at **state D** is the *EXIT ACTION* and gives the agent a **reward of  $x$** . The **reward for non-exit actions is always 1**.



- (a) Let all actions be deterministic. Assume  $\gamma = \frac{1}{2}$ . Express the following in terms of  $x$ .

$$V^*(D) = x$$

$$V^*(C) = \max(1 + 0.5x, 2)$$

$$V^*(A) = \max(1 + 0.5x, 2)$$

$$V^*(B) = \max(1 + 0.5(1 + 0.5x), 2)$$

The 2 comes from the utility being an infinite geometric sum of discounted reward =  $\frac{1}{(1-\frac{1}{2})} = 2$

- (b) Let any non-exit action be successful with probability =  $\frac{1}{2}$ . Otherwise, the agent stays in the same state with reward = 0. The *EXIT ACTION* from the **state D** is still deterministic and will always succeed. Assume that  $\gamma = \frac{1}{2}$ .

For which value of  $x$  does  $Q^*(A, \text{DOWN}) = Q^*(A, \text{RIGHT})$ ? Box your answer and justify/show your work.

$$Q^*(A, \text{DOWN}) = Q^*(A, \text{RIGHT}) \text{ implies } V^*(A) = Q^*(A, \text{DOWN}) = Q^*(A, \text{RIGHT})$$

$$V^*(A) = Q^*(A, \text{DOWN}) = \frac{1}{2}(0 + \frac{1}{2}V^*(A)) + \frac{1}{2}(1 + \frac{1}{2}x) = \frac{1}{2} + \frac{1}{4}(V^*(A)) + \frac{1}{4}x \quad (1)$$

$$V^*(A) = \frac{2}{3} + \frac{1}{3}x \quad (2)$$

$$V^*(A) = Q^*(A, \text{RIGHT}) = \frac{1}{2}(0 + \frac{1}{2}V^*(A)) + \frac{1}{2}(1 + \frac{1}{2}V^*(B)) = \frac{1}{2} + \frac{1}{4}V^*(A) + \frac{1}{4}V^*(B) \quad (3)$$

$$V^*(A) = \frac{2}{3} + \frac{1}{3}V^*(B) \quad (4)$$

Because  $Q^*(B, \text{LEFT})$  and  $Q^*(B, \text{DOWN})$  are symmetric decisions,  $V^*(B) = Q^*(B, \text{LEFT})$ .

$$V^*(B) = \frac{1}{2}(0 + \frac{1}{2}V^*(B)) + \frac{1}{2}(1 + \frac{1}{2}V^*(A)) = \frac{1}{2} + \frac{1}{4}V^*(B) + \frac{1}{4}V^*(A) \quad (5)$$

$$V^*(B) = \frac{2}{3} + \frac{1}{3}V^*(A) \quad (6)$$

Combining (2), (4), and (6) gives us:

$$x = 1 \tag{7}$$

There is also a shortcut which involves you noticing that the problem is highly symmetrical such that  $Q^*(A, DOWN) = Q^*(A, RIGHT)$  is the same as solving the equivalence of  $V^*(A)$  in the previous part and the utility of an infinite cycle with reward scaled by half (to account for staying) and discount = 0.5. That leads us to conclude  $0.5 + 0.5x = \frac{0.5}{1-0.5} = 1$  so  $x = 1$

- (c) We now add one more layer of complexity. Turns out that the reward function is not guaranteed to give a particular reward when the agent takes an action. Every time an agent transitions from one state to another, once the agent reaches the new state  $s'$ , a fair 6-sided dice is rolled. If the dices lands with value  $x$ , the agent receives the reward  $R(s, a, s') + x$ . The sides of dice have value 1, 2, 3, 4, 5 and 6.

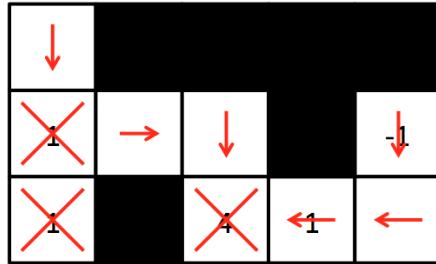
Write down the new bellman update equation for  $V_{k+1}(s)$  in terms of  $T(s, a, s')$ ,  $R(s, a, s')$ ,  $V_k(s')$ , and  $\gamma$ .

$$\begin{aligned} V_{k+1}(s) &= \max_a \sum_{s'} T(s, a, s') [\frac{1}{6} (\sum_{i=1}^6 R(s, a, s') + i) + \gamma V_k(s)] \\ &= \max_a \sum_{s'} T(s, a, s') (R(s, a, s') + 3.5 + \gamma V_k(s)) \end{aligned}$$

### Q3. MDPs: Reward Shaping

PacBot is in a Gridworld-like environment  $E$ . It moves deterministically Up, Down, Right, or Left, or at any time it can exit to a terminal state (where it remains). If PacBot is on a square with a number written on it, it receives a reward of that size **on Exiting**, and it receives a reward of 0 for Exiting on a blank square. Note that when it is on any of the squares (including numbered squares), it can either move Up, Down, Right, Left or Exit. However, it only receives a non-zero reward when it Exits on a numbered square.

- (a) Draw an arrow in **each** square (including numbered squares) in the following board to indicate the optimal policy PacBot will calculate with the discount factor  $\gamma = 0.5$ . (For example, if PacBot would move Down from the square in the middle, draw a down arrow in that square.) If PacBot's policy would be to exit from a particular square, draw an X in that square.



In order to speed up computation, Pacbot computes its optimal policy in a new environment  $E'$  with a different reward function  $R'(s, a, s')$ . If  $R(s, a, s')$  is the reward function in the original environment  $E$ , then  $R'(s, a, s') = R(s, a, s') + F(s, a, s')$  is the reward function in the new environment  $E'$ , where  $F(s, a, s') \in \mathbb{R}$  is an added “artificial” reward. If the artificial rewards are defined carefully, PacBot's policy will converge in fewer iterations in this new environment  $E'$ .

- (b) To decouple from the previous question's board configuration, let us consider that Pacbot is operating in the world shown below. Pacbot uses a function  $F$  defined so that  $F(s, a, s') = 10$  if  $s'$  is closer to C relative to  $s$ , and  $F(s, a, s') = 0$  otherwise (consider C to be closer to C than B or A). Let us also assume that the action space is now restricted to be between Right, Left, and Exit only.



Either left or right from B is correct.

In the diagram above, indicate by drawing an arrow or an X in each square, as in part (a), the optimal policy that PacBot will compute in the new environment  $E'$  using  $\gamma = 0.5$  and the modified reward function  $R'(s, a, s')$ .

- (c) PacBot's utility comes from the discounted sum of rewards **in the original environment**. What is PacBot's expected utility of following the policy computed above, starting in state A if  $\gamma = 0.5$ ? **0**
- (d) Find a non-zero value for  $x$  in the table showing  $F(s, a, s')$  drawn below, such that PacBot is guaranteed to compute an optimal policy that maximizes its expected true utility for **any** discount factor  $\gamma \in [0, 1)$ .

	Value
$F(A, \text{Right}, B)$	10
$F(B, \text{Left}, A)$	$x$
$F(B, \text{Right}, C)$	10
$F(C, \text{Left}, B)$	$x$

$x =$  Any number less than  $-10$  will also work. No other solution is correct.