Toolbox I.A. - ICM2A – ENSM-SE
Reinforcement Learning
Exercises

(from « Reinforcement Learning : An Introduction », R.S.Sutton & A.G.Barto, MIT Press, 1998)

1. Devise three example tasks of your own that fit into the reinforcement learning framework, identifying for each its states, actions, and rewards. Make the three examples as *different* from each other as possible. The framework is abstract and flexible and can be applied in many different ways. Stretch its limits in some way in at least one of your examples.

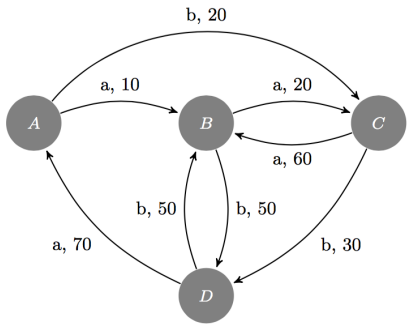
2. Is the reinforcement learning framework adequate to usefully represent *all* goal-directed learning tasks? Can you think of any clear exceptions?

3. Imagine that you are designing a robot to run a maze. You decide to give it a reward of +1 for escaping from the maze and a reward of zero at all other times. The task seems to break down naturally into episodes--the successive runs through the maze – so you decide to treat it as an episodic task, where the goal is to maximize expected total reward :
$$R_t = r_{t+1} + r_{t+2} + r_{t+3} + \cdots + r_T,$$
After running the learning agent for a while, you find that it is showing no improvement in escaping from the maze. What is going wrong? Have you effectively communicated to the agent what you want it to achieve?

4. Imagine that you are a vision system. When you are first turned on for the day, an image floods into your camera. You can see lots of things, but not all things. You can't see objects that are occluded, and of course you can't see objects that are behind you. After seeing that first scene, do you have access to the Markov state of the environment? Suppose your camera was broken that day and you received no images at all, all day. Would you have access to the Markov state then?

From J.B. Alonso, Universitat Politècnica de Catalunya :

We want to build a system able to control a process with four states {A,B,C,D} were we can perform the actions a and b. The following figure shows the state transition function (δ) and the reinforcement (r) obtained by each action:



Using the Q-Learning algorithm with γ = 0.9 and α = 1, the action value function Q that is obtained with the sequence that begins in the state A and performs the actions {a,a,b,a,b,a} is:

(a)

Q(s,a)	A	B	C	D
a	10	20	30	70
b	20	0	60	0

(b)

Q(s,a)	A	B	C	D
a	10	20	78	79
b	47	0	30	0

(c)

Q(s,a)	A	B	C	D
a	10	20	30	70
b	20	50	60	50

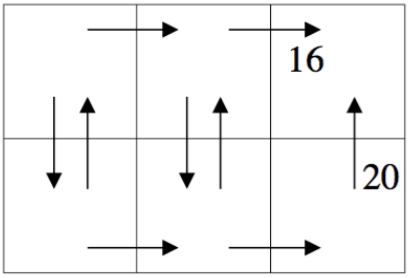
(d)

Q(s,a)	A	B	C	D
a	78	10	47	30
b	79	0	20	0

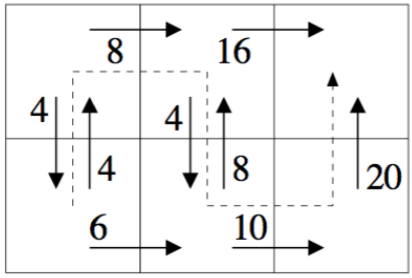
From V. Lesser, University of Massachusetts Amherst :

Consider the deterministic world below (part (a)). Allowable moves are shown by arrows, and the numbers indicate the reward for performing each action. If there is no number, the reward is zero.

Given the Q values in (b), show the changes in the Q estimates when the agent take the path shown by the dotted line (the agent starts in the lower left cell) when $\gamma = 0.5$. Show all of your work.



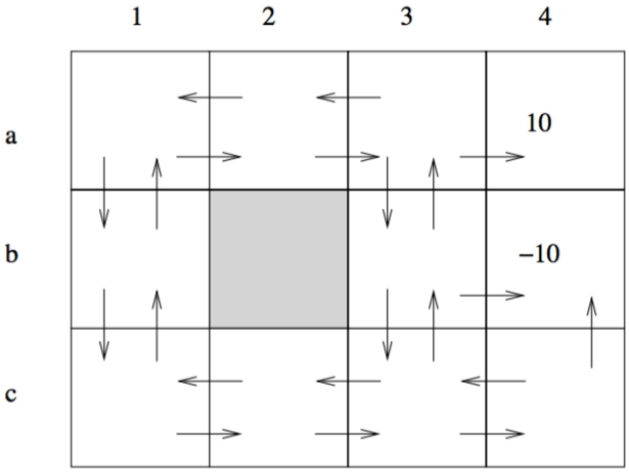
(a)



(b)

From Doug Aberdeen, Australian National University :

Simulate Q-learning for a robot walking around in the following environment (b2 is a wall, entering b4 gives a penalty of -10, entering a4 gives a reward of 10).



Indicate Q-values after the following episodes, using the "back-propagated" Q update rule (i.e. after getting in a goal state, updating the Q values in reverse order from goal to start, $\gamma = 0.9$).

1) a1,a2,a3,b3,b4

2) c2,c1,b1,a1,a2,a3,a4

3) c4,c3,b3,a3,a4

Assume the robot will now use the policy of always performing the action having the greatest Q value. Indicate this policy on the drawing. Is it optimal?