Reinforcement Learning

Exercise 1

Submitted by:
Mostafa Hussein
Miray Yuce
Arlette Perez

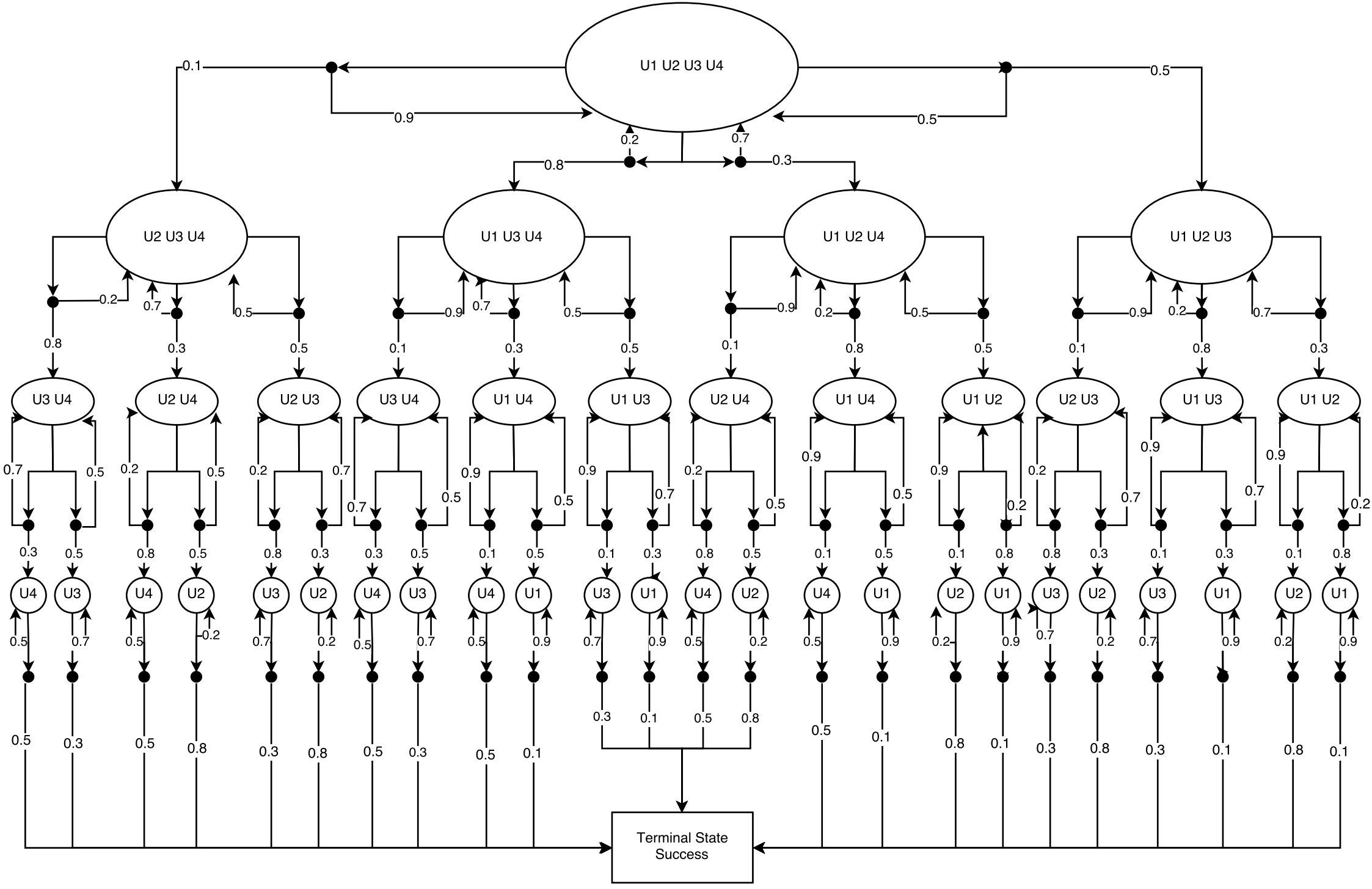
1 Markov Decision Processes

(a) Formalize the above described problem as a Markov Decision Process. You do not have to write out the individual elements completely (i.e. name all states explicitly or provide a full transition matrix), but define the contents in unambiguous expressions

MDP = <S,A,P,R,γ>
S = {from 0000 to 1111 | where each digit means if a task was successful, or not}
A = {u1, u2, u3, u4}

An example of states starting from 0000, possible next states, actions, transitions and rewards can be seen below:

s	s’	a	P	R
0000	0001	U4	0.5	2
	0000	U4	0.5	0
	0010	U3	0.3	3
	0000	U3	0.7	0
	0100	U2	0.8	1
	0000	U2	0.2	0
	1000	U1	0.1	4
	0000	U1	0.9	0


U1 U2 U3 U4
U2 U3 U4
U1 U3 U4
U1 U2 U4
U1 U2 U3
U3 U4
U2 U4
U2 U3
U3 U4
U1 U4
U1 U3
U2 U4
U1 U4
U1 U2
U2 U3
U1 U3
U1 U2
U4
U3
U4
U2
U3
U2
U4
U1
U3
U1
U4
U2
U4
U1
U2
U1
U3
U2
U3
U1
U2
U1
Terminal State
Success

(b) How would you model the risk of failing the exam in this scenario? *Hint: You can introduce terminal rewards.*

- Number of failing states: 5
- Number of success states: $2^4 - 5 = 11$

	Failing states		
u	u2	u3	u4
1			
0	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1
0	1	0	1

Failing terminal node reward is equal to -10 so that the agent will avoid getting there, while all other success terminal states have 0 reward

(c) Compare both policies by determining the path costs for both policies.

We wrote a code (see RL_ex1.py) and from the obtained results, we could conclude that policy A is better than policy B. The state-value function of A gives us a reward of positive values and higher than the value function of B. In the case of B, most of the cases the value function is negative, since the order of the questions is in increasing probability of success. So if the student S fails at u1, she will try again and again, until u1 is successful, on the contrary for A, the order is in decreasing probability of success, it is easier to move further questions.

We calculate the value function by using the formula:

$$V_\pi = \sum_a \pi(s,a) \sum_{s'} P^a_{ss'} \left[R^a_{ss'} + \gamma V_\pi(s') \right] = \sum_a \pi(s,a) \sum_{s'} P^a_{ss'} \left[R^a_{ss'} + \gamma E_\pi \left[\sum_{k=0}^{4} \gamma^k r_{t+k+2} \vee s_{t+1} = s' \right] \right] = ¿$$

p_s * (r + v_n) + (1-p_s) * v_c

where p_s: *probability of the state*

r: *reward*

v_n: *value function of the next state*

v_c: *value function of the current state*

Output of our script:

Value of policy a: 1.8

Value of policy b: 0.4

(d) In what order does this strategy choose the tasks?

It seems A is a very good policy for a non-stationary policy. But if we can have a stationary policy, then we don't have to check if we are end of the exam, which means N = 5. In that case, we can try the actions infinite times. Since u2 has a low reward, we can start with u1, then we can change to an action with higher probability which is u4, and so on. At the end, the order of the questions is: u1, u4, u 2, u3 for policy C.

Output of our script:

Value of policy c: 3.1

(e) Suggest a method that reduces the probability of failing for each of the above policies. You can describe in non-formal sentences. Hint: Use non-stationary policies.

Simply, a non-stationary Markov chain, if the transition probabilities change with time and a stationary Markov chain if the transition probabilities are independent of time. So non-stationary MDPs can be viewed as a special case of countable-state stationary MDPs by appending the states with a time-index.
Nonstationary policies are functions from states and times to actions:-
- π:S x T $\rightarrow$ A, where T is the non-negative integers
- π(s,t) tells us what action to take at state s when there are t stages-to-go

So in order to reduce the probability of failing in the exam, a student must be considering the time series, or to be clear, how much time is left for him to reach his terminal state, and accordingly, he will not be working with a random policy, where he tries to solve a random question. However, he will consider the number of trials left -because time matters-, so he can change his transition state according to this. For simplicity, in the previous question, each try was given the same time, but in non-stationary problems, this can change. Trials could not be having the same time, so a trial in a difficult question, can take more time than a trial in an easy question.
For example, he will start of solving the question which has the highest grade and allows him to success in the exam by just achieving it. In this case, U1 gives us the greatest number of points (4), which is actually the passing grade (40%). In case he failed in solving it for 3 times for example, so he must not to choose randomly what to solve next, because he now knows that only 2 trials left (time limted). So for example if in these 2 trials, he solved U2 (1 point), and U4 (2 points), he will achieve a total score of 3 points which is a failure terminal state. So in this case, he must try to solve U3 before, because if he can do it from 1 trial, he will already get 3 reward points, then in his last trial he can go with U2.

To summarize, taking the time into consideration will allow the agent to know when he is about to end/reach a terminal state, and then he will consider then which actions it has to follow or do next, and the transition states will be changed according to this.

(f) How good must student T be prepared for the topic of task 1?

Student T must have p_1 >= 50% at least to beat student S with policy A

Output of our script:

Value of student t: 2.4

2 Bellman Equation

(a) Show exemplary for state s3;3 in the middle of the grid with v_(s3;3) = 0:7 that the Bellman equation is satisfied for all neighbouring states.

		2.3		
	0.7	0.7	0.4	
		-0.4		

$$V_\pi = 0.7 = \sum_a \pi(s,a) \sum_{s'} P_s^a \left[R_{ss'}^a + \gamma V_\pi(s')\right]$$

$$= 0.25[0.9(2.3)] + 0.25[0.9(0.7)] + 0.25[0.9(-0.4)] + 0.25[0.9(0.4)]$$
$$= 0.675 \approx 0.7$$

(b) Explain why the value of state B is higher than the direct reward. Why does this not hold for state A?

The main reason is that the value of a state depends on the neighbouring states. We observed from the table that the neighbours of B have low utilities; however, the neighbours of A have higher utilities

(c) How does the optimal policy change? What are the new values of states A and B?

The values of A and B before the changing are:

$$v_A^\pi = 10 + 0.9 * (-1.3) = 8.83$$
$$v_B^\pi = 5 + 0.9 * (0.4) = 5.36$$

The value of A after the changing is: $v_A^\pi = 4 + 0.9 * (-1.3) = 2.83$

$$v_B^\pi = 5 + 0.9 * (0.4) = 5.36$$