

### 3 MDP (12 points)

1. (3 pts) In MDPs, the values of states are related by the Bellman equation:

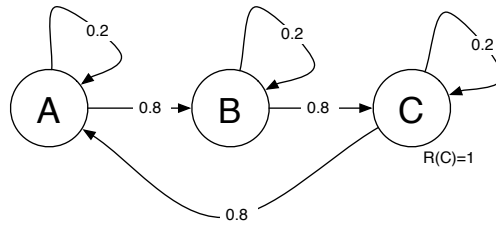
$$U(s) = R(s) + \gamma \max_a \sum_{s'} P(s'|s, a)U(s')$$

where  $R(s)$  is the reward associated with being in state  $s$ . Suppose now we wish the reward to depend on actions; i.e.  $R(a, s)$  is the reward for doing  $a$  in state  $s$ . How should the Bellman equation be rewritten to use  $R(a, s)$  instead of  $R(s)$ ?

2. (9 pts) Can any search problem with a finite number of states be translated into a Markov decision problem, such that an optimal solution of the latter is also an optimal solution of the former? If so, explain precisely how to translate the problem AND how to translate the solution back; illustrate your answer on a 3 state search problem of your own choosing. If not give a counterexample.

## 4 Reinforcement Learning (13 points)

Consider an MDP with three states, called A, B and C, arranged in a loop.



There are two actions available in each state:

- *Move<sub>s</sub>*: with probability 0.8, moves to the next state in the loop and with probability 0.2, stays in the same state.
- *Stay<sub>s</sub>*: with probability 1.0 stays in the state.

There is a reward of 1 in state C and zero reward elsewhere. The agent starts in state A. Assume that the discount factor is 0.9, that is,  $\gamma = 0.9$ .

1. (6 pts) Show the values of  $Q(a, s)$  for 3 iterations of the TD Q-learning algorithm (equation 21.8 in Russell & Norvig):

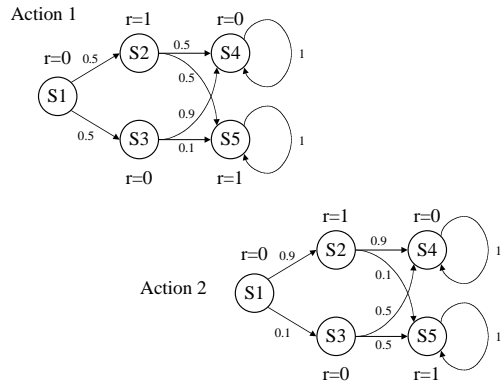
$$Q(a, s) \leftarrow Q(a, s) + \alpha(R(s) + \gamma \max_{a'} Q(a', s') - Q(a, s))$$

Let  $\alpha = 1$ , note the simplification that follows from this. Assume we always pick the Move action and end up moving to the adjacent state. That is, we see a state-action sequence: *A, Move, B, Move, C, Move, A*. The Q values start out as 0.

	iter=0	iter=1	iter=2	iter=3
Q(Move,A)	0			
Q(Stay,A)	0			
Q(Move,B)	0			
Q(Stay,B)	0			
Q(Move,C)	0			
Q(Stay,C)	0			



## 14 Markov Decision Processes



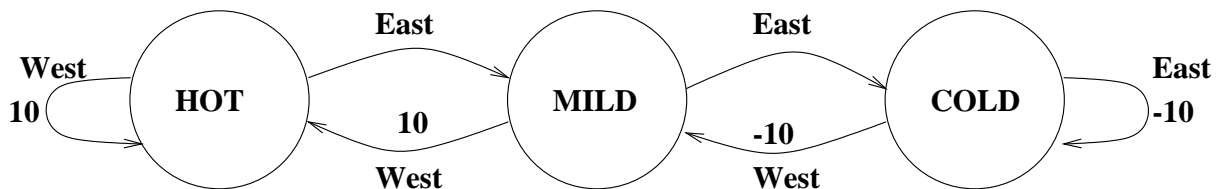
(10) What are the values of the states in the following MDP, assuming  $\gamma = 0.9$ ? In order to keep the diagram from being too complicated, we've drawn the transition probabilities for action 1 in one figure and the transition probabilities for action 2 in another. The rewards for the states are the same in both cases.

### Problem 3 Reinforcement Learning (24 points)

This problem has two parts, one focused on Q-learning in a deterministic world and one focused on Q-learning in a nondeterministic world. If you get stuck, be sure to scan the remaining parts, because some latter parts can be answered readily without answering previous parts.

#### Deterministic world

Consider the simple 3-state deterministic weather world sketched in the figure below. There are two actions for each state, namely moving West and East. The non-zero rewards are marked on the transition edges, hence edges without a number correspond to a zero reward.



#### Part A (4 points)

A robot starts in the state Mild. It is actively learning its  $\hat{Q}$ -table and moves in the world for 4 steps choosing actions according to a toss of a coin, namely **West, East, East, West**. The initial values of its Q-table are 0 and the discount factor is  $\gamma = 0.5$ . Fill in the Q-values for the sequence of states visited.

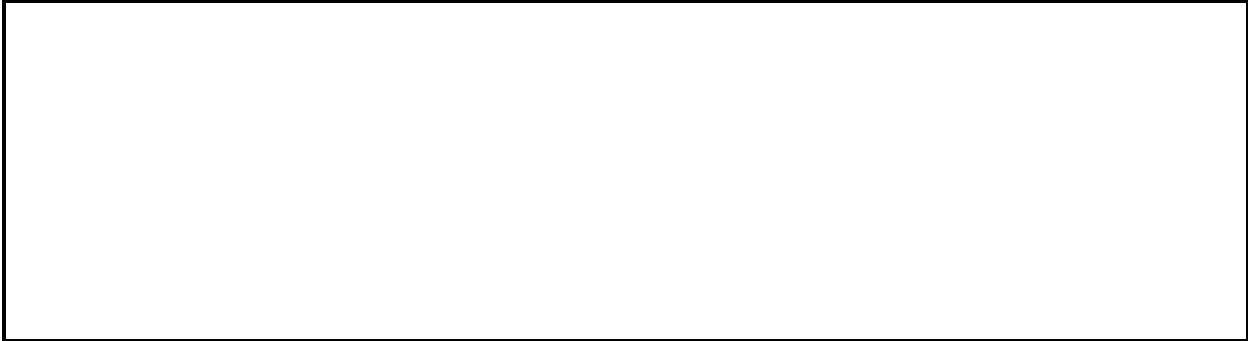
	Initial State: MILD		Action: West New State: HOT		Action: East New State: MILD		Action: East New State: COLD		Action: West New State: MILD	
	East	West	East	West	East	West	East	West	East	West
HOT	0	0								
MILD	0	0								
COLD	0	0								

#### Part B (2 points)

How many possible **policies** are there in this 3-state deterministic world? .

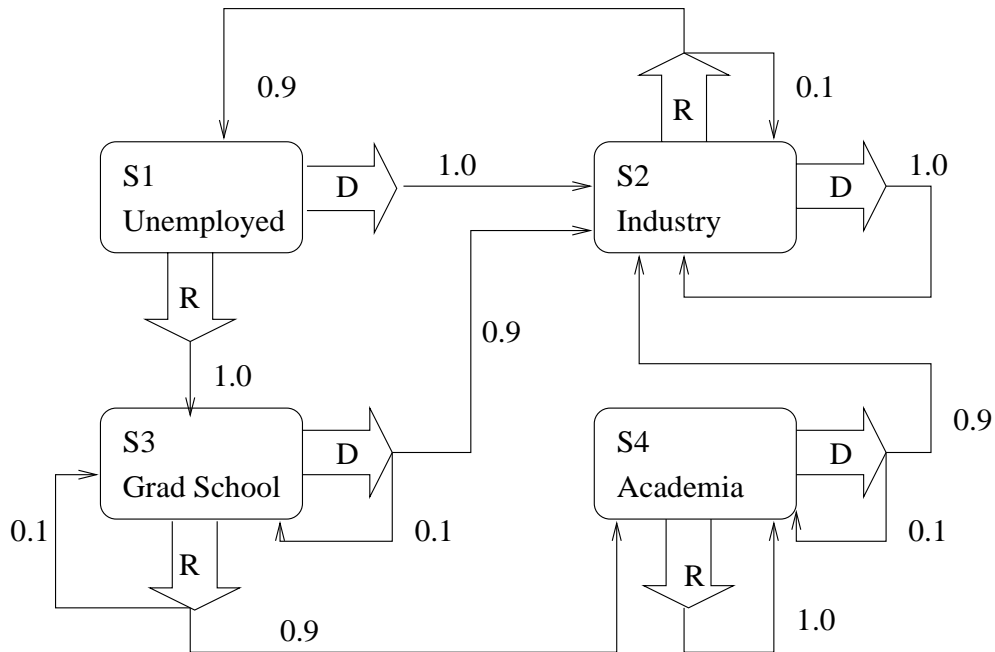
### Part C (4 points)

Why is the policy  $\pi(s) = \text{West}$ , for all states, *better than* the policy  $\pi(s) = \text{East}$ , for all states?



### Nondeterministic world

Consider the Markov Decision Process below. Actions have nondeterministic effects, i.e., taking an action in a state always leads to one next state, but which state is the one next state is determined by transition probabilities. These transition probabilities are shown in the figure attached to the transition arrows from states and actions to states. There are two actions out of each state: **D** for development and **R** for research.



## Part D (6 points)

Consider the following deterministic *ultimately-care-only-about-money* reward for any transition *starting* at a state:

REWARD			
S1	S2	S3	S4
0	100	0	10

Let  $\pi^*$  represent the optimal policy which is *given* to you, namely, for  $\gamma = 0.9$ ,  $\pi^*(s) = D$ , for any  $s = S1, S2, S3, \text{ or } S4$ .

### D.1

Compute the optimal expected reward from each state, namely  $V^*(S1)$ ,  $V^*(S2)$ ,  $V^*(S3)$ ,  $V^*(S4)$  according to this policy. *Hints: Start by calculating  $V^*(S2)$ . Remember the nondeterministic transitions.*

### D.2

What is the Q-value,  $Q(S2,R)$ ?

## Part E (4 points)

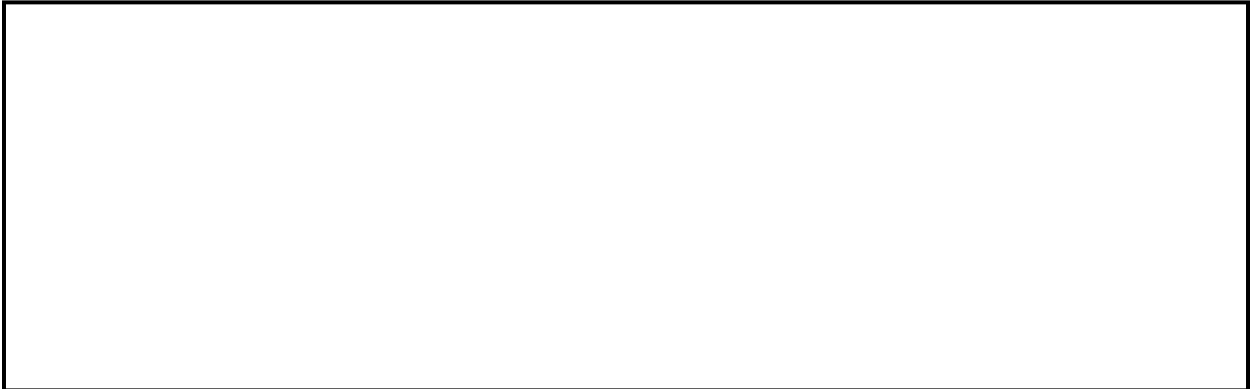
### E.1

What is the problem with a Q-learning agent that always takes the action whose estimated  $Q$ -value is currently the highest? (This is like the max-first exploration strategy in the 4th programming assignment.)



### E.2

What is the problem with a Q-learning agent that ignores its current estimates of  $Q$  in order to explore everywhere? (This is like the random exploration strategy in the 4th programming assignment.)





## Part F (6 points)

Consider the Q-learning training rule for a nondeterministic Markov Decision Process:

$$\hat{Q}_n(s, a) \leftarrow (1 - \alpha_n)\hat{Q}_{n-1}(s, a) + \alpha_n[r + \gamma \max_{a'} \hat{Q}_{n-1}(s', a')],$$

where  $\alpha_n = \frac{1}{1 + \text{visits}_n(s, a)}$ , and  $\text{visits}_n(s, a)$  is the number of times that the transition  $s, a$  was visited at iteration  $n$ .

Answer with True (T) or False (F):

- $\alpha_n$  decreases as the number of times the learner visits the transition  $s, a$  increases.
- The weighted sum through  $\alpha_n$  makes the Q-values oscillate as a function of the nondeterministic transitions and therefore not converge.
- If the world is deterministic, the Q-learning training rule given above converges to the same as the specific one for deterministic worlds.