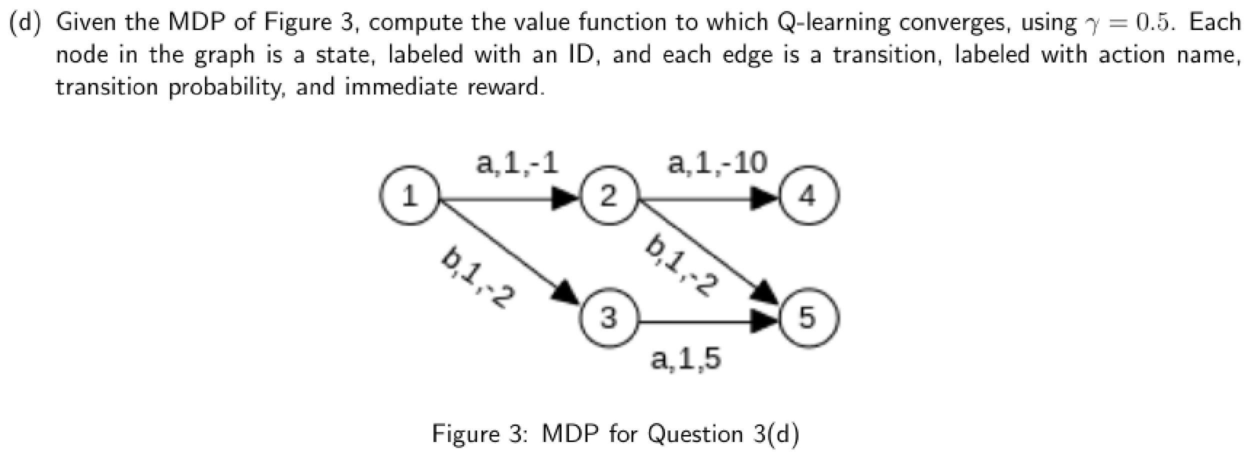
## From Jan 2020

1.



(d) Given the MDP of Figure 3, compute the value function to which Q-learning converges, using $\gamma = 0.5$. Each node in the graph is a state, labeled with an ID, and each edge is a transition, labeled with action name, transition probability, and immediate reward.

Figure 3: MDP for Question 3(d)

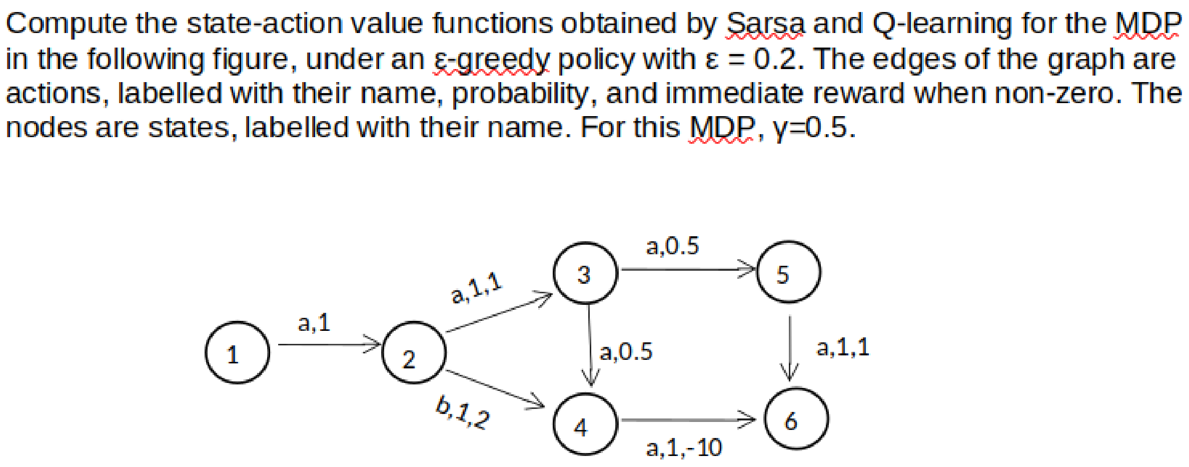
## From Jan 2018

2. A robot can execute the following actions: *goto(R)*, which moves it to room R; *pickup*, which picks up a cup of coffee; *putdown*, which puts down the cup. The action *pickup* can only be executed if the cup is in the same room as the robot, while the action *putdown* can only be executed if the robot is holding the cup. The robot can perceive in which room it is, where objects are in the room, and whether or not there is a person in the room. It receives a reward of +1000 for putting down a cup of coffee in room A when there is a person in it, and a reward of 0 for every other action. Consider the representation for the state space in which each state is a vector of the following values: <Room, CupInRoom, PersonInRoom, HoldingCup>. For instance, a possible state is <A,True,True,False>. Is this representation Markovian? Why? If you think it is Markovian, propose a Non-Markovian alternative, while if you think it is not Markovian, then propose a Markovian one.

## From Jan 2019

3. Describe the difference between an on-policy and off-policy reinforcement learning algorithm. Provide an example of each method, and an MDP where they converge to different value functions.

4.



Compute the state-action value functions obtained by Sarsa and Q-learning for the MDP in the following figure, under an ε-greedy policy with ε = 0.2. The edges of the graph are actions, labelled with their name, probability, and immediate reward when non-zero. The nodes are states, labelled with their name. For this MDP, γ=0.5.