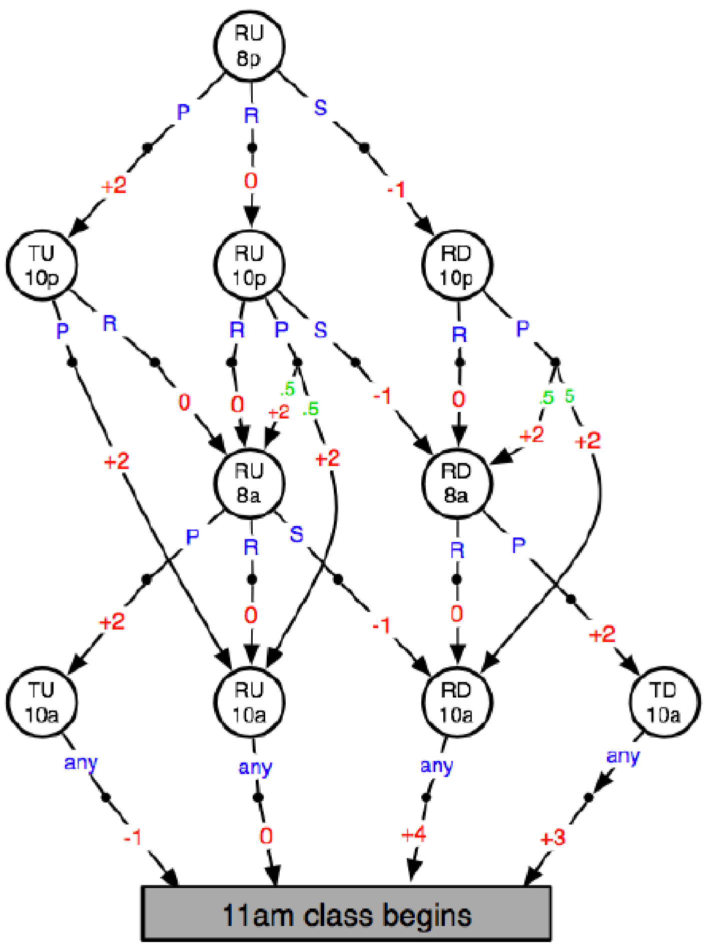
CS 6362: Machine Learning
Assignment 3
Reinforcement Learning: The Party Problem

Due: 11/01/2019, midnight

Total: 100 Points

The figure below describes a Markov decision process based on life as a student and the decisions one must make to both have a good time and remain in good academic standing.



Interpretation of MDP nodes and links:

T = Tired

D = home work Done

U = homework Undone (not done)

8p = eight o'clock pm (and so on, 10p; 8a, 10a – all denote time)

Actions:

P = Party

R = Rest

S = Study

any means any action has the same effect

Note not all actions are possible in all state s

Red numbers are rewards

Green numbers are transition probabilities (all those not labeled are probability 1.0)

Programming Assignment 3a: Party Problem Programming (40 points)

Implement a program that models the Party Problem described above. Use any programming language of your choice. Assume that the agent follows a random equiprobable policy (i.e. the probability of picking a particular action while in a given state is equal to 1 / number of actions that can be performed from that state).

Run your program for 50 episodes. For each episode, have your program print out the agent's sequence of experience (i.e., the ordered sequence of states/actions/rewards that occur in the episode) as well as the sum of the rewards received in that episode (i.e., the Return with respect to the start state) in a readable form.

What you need to Hand In (as a pdf report + in electronic executable form):

- The sequences of experience from each episode, including the Return observed in that episode. (pdf report)
- The values of each state (hint: use the Bellman equations). (pdf report)
- The average Return from the fifty episodes. (pdf report)
- The source code of your program. (electronic form + instructions for executing your program)

Programming Assignment 3b: Party Problem Programming – Policy Iteration (60 Points)

Implement the policy iteration algorithm (see slides 15-16, Lecture 14) to learn the optimal policy for the Party Problem described above. Set the initial policy to "Rock & Roll all night and Party every day" (i.e. policy should choose to party regardless of what state the agent is in). Perform each policy evaluation step until the largest change in a state's value (Δ in slide 16) is smaller than 0.001 (θ in the algorithm on slide 16). Print out the policy and the value function for each iteration (policy change) of the algorithm in a readable form.

What to Hand In (as a pdf report + in electronic executable form):

The policy and value function for each iteration of the algorithm. (pdf report)

The source code of your program (electronic form + documentation for executing your program)