This handout includes space for every question that requires a written response. Please feel free to use it to handwrite your solutions (legibly, please). If you choose to typeset your solutions, the `README.md` for this assignment includes instructions to regenerate this handout with your typeset LaTeX solutions.

1.a

(2.14)

Via value iteration

$$V_{opt}(s) = \begin{cases} 0 & (\text{if } \text{isEnd}(s)) \\ \max_{a \in \text{Actions}(s)} Q_{opt}(s,a) & \end{cases}$$

$$Q_{opt}(s,a) = \sum_{s'} T(s,a,s')\left[r + \gamma V_{opt}^{(t-1)}(s')\right]$$

$\gamma = 1$

$$r = \begin{cases} -5 & \\ 20 & s=-2 \\ 100 & s=2 \end{cases}$$

	state s				
	-2	-1	0	1	2
$V_{opt}^{(0)}(s)$	0	0	0	0	0
$V_{opt}^{(1)}(s)$	0	15	-5	26.5	0
$V_{opt}^{(2)}(s)$	0	14	13.45	23	0
$\pi_{opt}(s)$	-	-1	+1	+1	-

1.a

1.b

s	a	s'	T(s,a,s')	r
0	+1	1	.3	-5
0	+1	-1	.7	-5
0	-1	1	.2	-5
0	-1	-1	.8	-5
1	+1	2	.3	100
1	+1	0	.7	-5
1	-1	2	.2	100
1	-1	0	.8	-5
-1	+1	-2	.7	20
-1	+1	0	.3	-5
-1	-1	-2	.8	20
-1	-1	0	.2	-5

$\gamma = 1$

$\text{Actions}(s) = \{-1, +1\}$

$$V_{opt}^{(t)}(s) = \max_{a \in \text{Actions}(s)} \left[\sum_{s'} T(s,a,s') \left[r + \gamma V_{opt}^{(t+1)}(s') \right] \right]$$

keep the action options

keep $\sum_{s'} T(s,a,s') = 1$ for $\sum_{s'}$

$$V_{opt}^{(1)}(0) = .3\left[-5 + (1)(0)\right] + .7\left[-5 + (1)(0)\right]$$

$$= -5$$

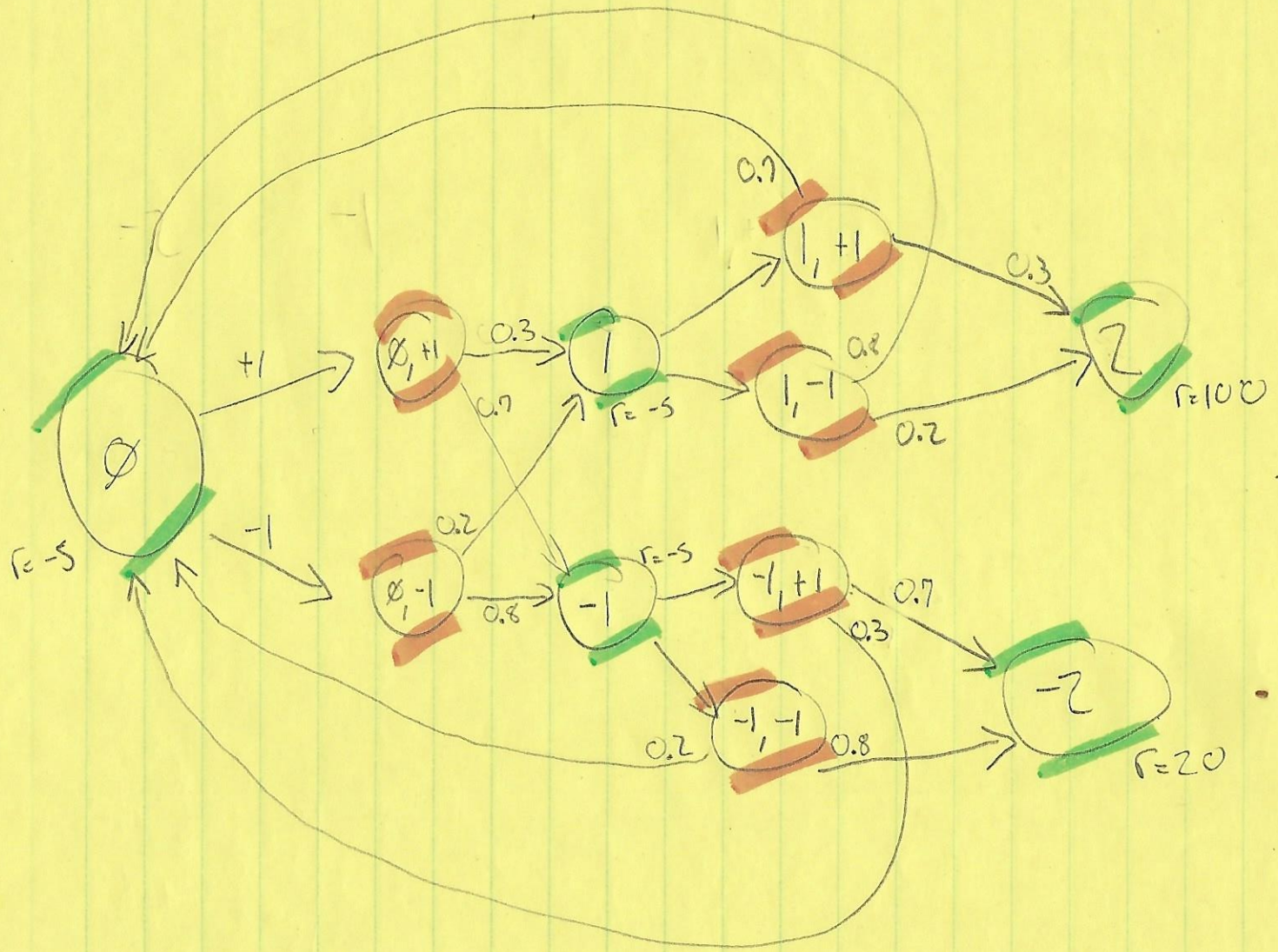
$$V_{opt}^{(2)}(0) = .3\left[-5 + (1)(2.5)\right] + .7\left[-5 + (1)(15)\right]$$

$$V_{opt}^{(1)}(1) = .3\left[100 + (1)(0)\right] + .7\left[-5 + (1)(0)\right]$$

$$V_{opt}^{(2)}(1) = .3\left[100 + (1)(0)\right] + .7\left[-5 + (1)(-5)\right]$$

$$V_{opt}^{(1)}(-1) = .8\left[20 + (1)(0)\right] + .2\left[-5 + (1)(0)\right]$$

See
214
216


∅
r=-5
+1
∅, +1
0.3
0.7
1
r=-5
1, +1
0.7
0.3
1, -1
0.8
0.2
2
r=100
-1
∅, -1
0.2
0.8
-1
r=-5
-1, +1
0.7
0.3
-1, -1
0.2
0.8
-2
r=20

2.a

MDP graph needs to be acyclic because convergence can't occur otherwise with $\lambda' = 1$

Or

we can introduce convergence with the introduction of a new end state that any of the prior states can end at. to integrate this the previous transition probabilities need to be slightly diminished to make room for the new state since the sum of T between s and s' must equal 1 one approach to do this and include the original λ of the original MDP is:

$\lambda' = 1$ $\quad A'(s) = A(s)$ $\quad S' = S \cup \{0\}$ $\quad 0$ = new end state

$R'(s,a,s') = R(s,a,s')$ since 0 is a end state we can make $R(s,a,0) = R_{opt}(s,a,s')$ for s or their reward for the optimal action for each state

$T'(s,a,s') = T(s,a,s') * \lambda \cup T(s,a,0) = (1-\lambda)$

4.b

For the small MDP our total reward for each of the RL iterations is binary in nature being either the optimal value or 0 .there is too much exploitation over exploration, resulting in a fairly minimal amount of the state space actually being effectively traversed. Not many of the weights were actually set despite a substantive amount of iterations. Which results in the MDP failing in larger statespace because they never reach the target state. Without some exploitation the MDP can also get lost, both value iteration and Q-learning broke in the larger state space. To fix this we can use epsilon-greedy or function approximation so that we can interpolate between these two extremes.

4.d

5.a

5.b

5.c

5.d