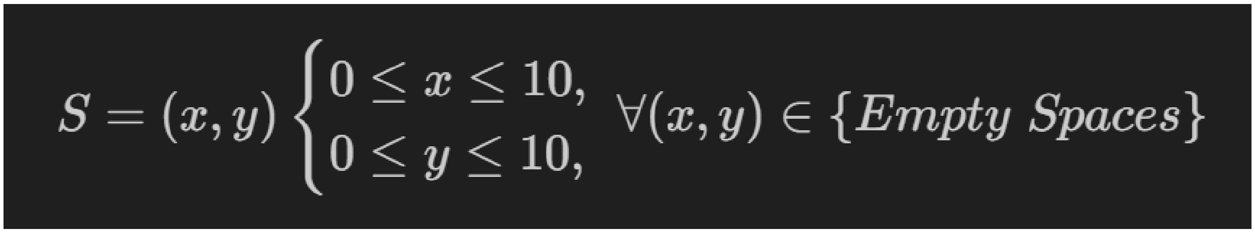
Adnan Amir

CS 5180 – HW2

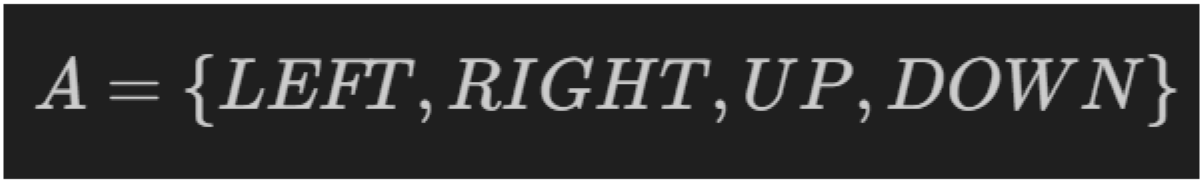
Q1 – Formulating an MDP

a. The state space consists of all empty spaces in the four rooms environment. It is an 11 x 11 grid so it has 121 states, but only 104 states are reachable, the others are obstacles. Thus, the state space is



$$S = (x, y) \begin{cases} 0 \leq x \leq 10, \\ 0 \leq y \leq 10, \end{cases} \forall (x, y) \in \{Empty\ Spaces\}$$

The action space is the 4 set of actions the agent can take

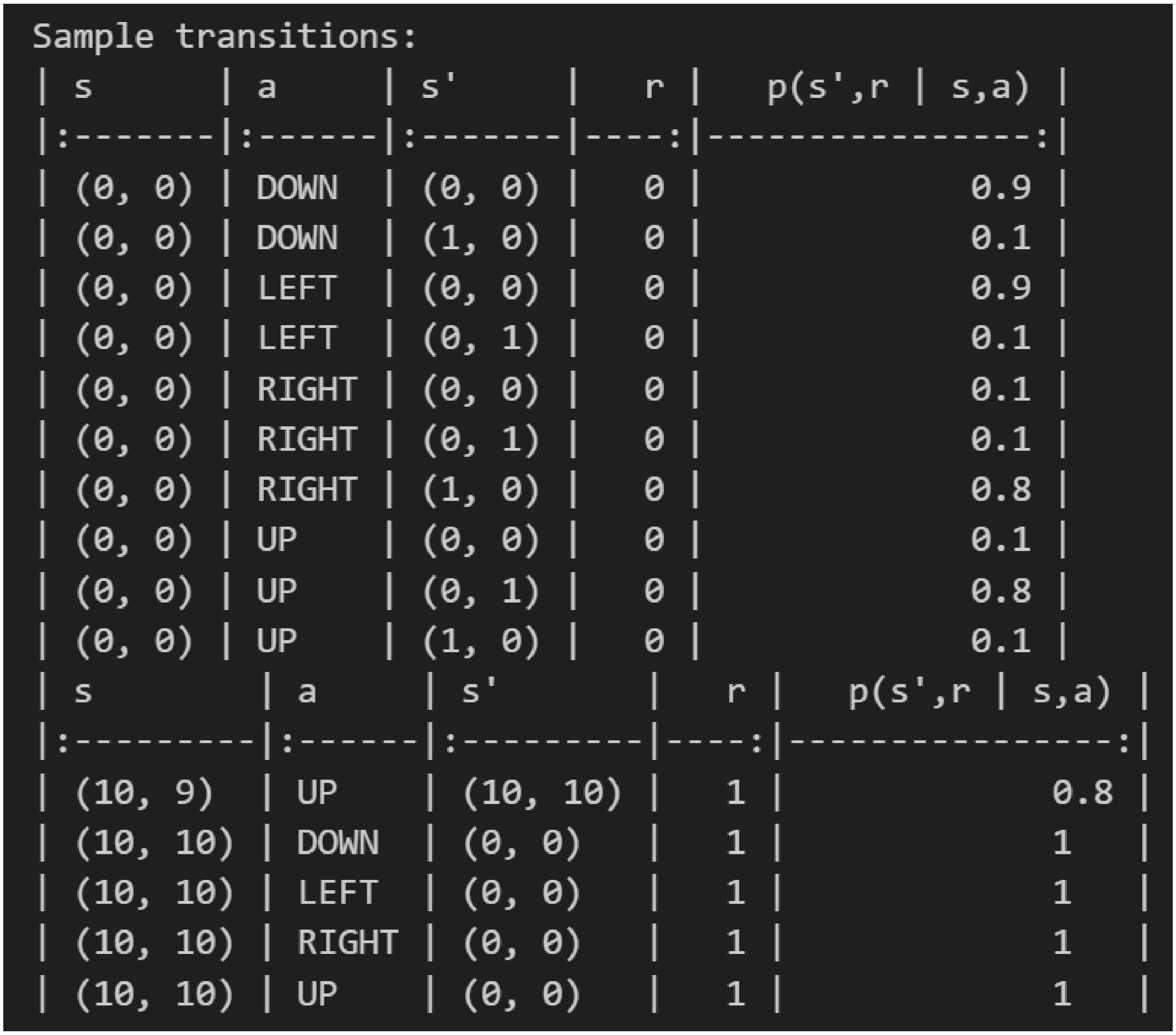


$$A = \{LEFT, RIGHT, UP, DOWN\}$$

b. There are 104 valid states, each of those states can have 4 possible actions. So 104 x 4 =416 entries. However, the transitions are not deterministic, for each action selected, there are 3 possibilities, so the total goes up to 416 x 3 = **1248 entries possible.**

The actual table will possibly have less entries than that because the goal state is not considered and there will be some identical entries on the walls.

c. Please see the csv attached for full output. The first 10 and the last 5 entries are attached below.

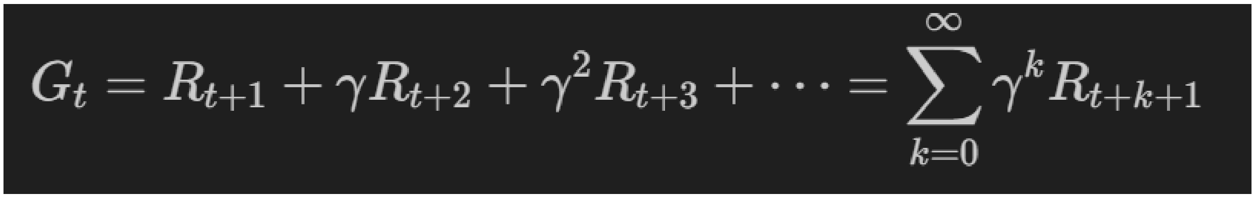


```
Sample transitions:
| s      | a     | s'     |   r |   p(s',r | s,a) |
|:-------|:------|:-------|----:|----------------:|
| (0, 0) | DOWN  | (0, 0) |   0 |             0.9 |
| (0, 0) | DOWN  | (1, 0) |   0 |             0.1 |
| (0, 0) | LEFT  | (0, 0) |   0 |             0.9 |
| (0, 0) | LEFT  | (0, 1) |   0 |             0.1 |
| (0, 0) | RIGHT | (0, 0) |   0 |             0.1 |
| (0, 0) | RIGHT | (0, 1) |   0 |             0.1 |
| (0, 0) | RIGHT | (1, 0) |   0 |             0.8 |
| (0, 0) | UP    | (0, 0) |   0 |             0.1 |
| (0, 0) | UP    | (0, 1) |   0 |             0.8 |
| (0, 0) | UP    | (1, 0) |   0 |             0.1 |
| s        | a     | s'       |   r |   p(s',r | s,a) |
|:---------|:------|:---------|----:|----------------:|
| (10, 9)  | UP    | (10, 10) |   1 |             0.8 |
| (10, 10) | DOWN  | (0, 0)   |   1 |             1   |
| (10, 10) | LEFT  | (0, 0)   |   1 |             1   |
| (10, 10) | RIGHT | (0, 0)   |   1 |             1   |
| (10, 10) | UP    | (0, 0)   |   1 |             1   |
```

Q2 – Discounted Return

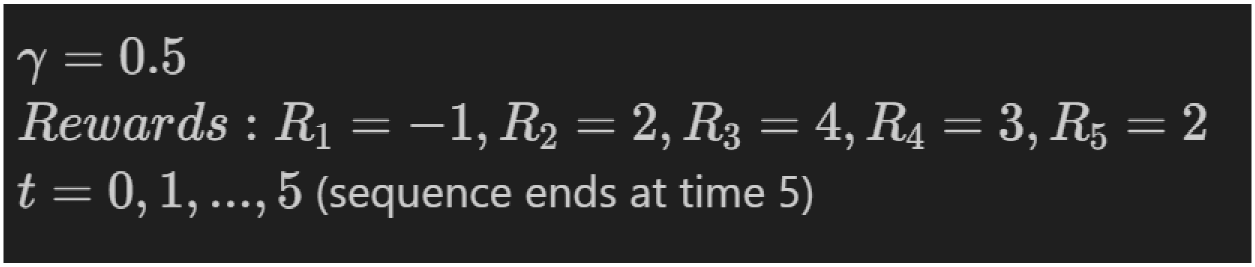
a. The discounted return is given by

Adnan Amir



$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

Given:



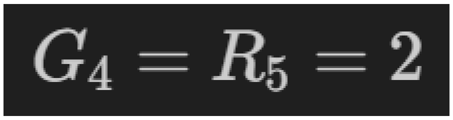
$\gamma = 0.5$
$Rewards : R_1 = -1, R_2 = 2, R_3 = 4, R_4 = 3, R_5 = 2$
$t = 0, 1, ..., 5$ (sequence ends at time 5)

Using the formula

G5 =

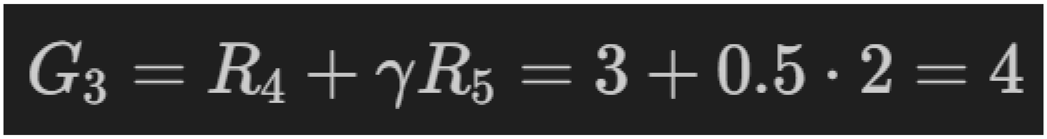
$$G_5 = 0 \quad (\text{no rewards after } R_5)$$

G4=



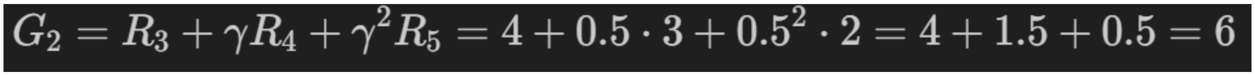
$$G_4 = R_5 = 2$$

G3=



$$G_3 = R_4 + \gamma R_5 = 3 + 0.5 \cdot 2 = 4$$

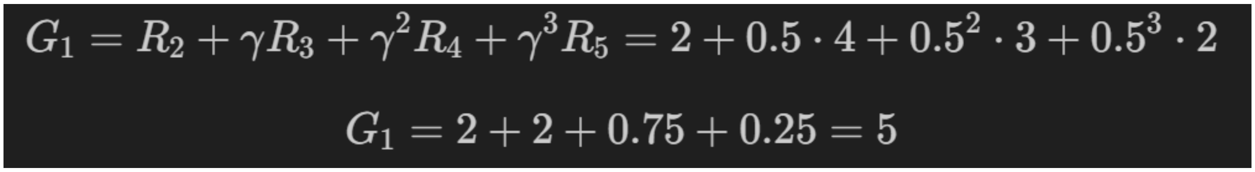
G2=



$$G_2 = R_3 + \gamma R_4 + \gamma^2 R_5 = 4 + 0.5 \cdot 3 + 0.5^2 \cdot 2 = 4 + 1.5 + 0.5 = 6$$

G1=

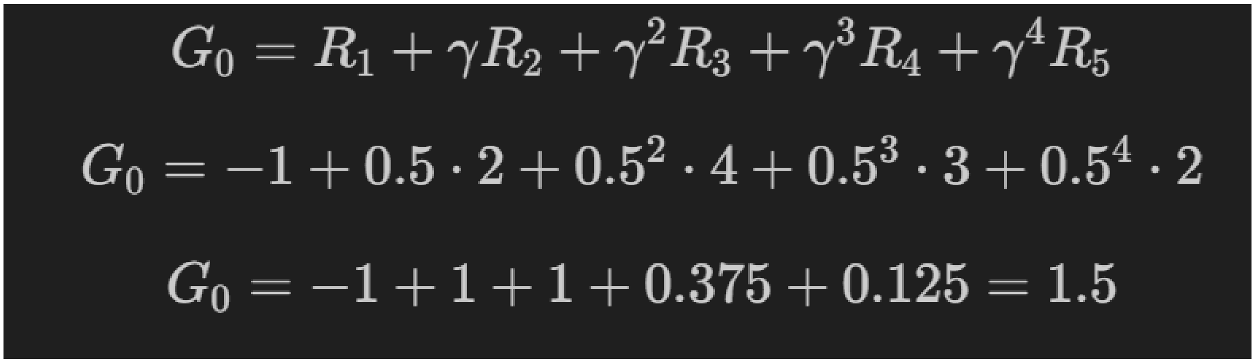
Adnan Amir



$$G_1 = R_2 + \gamma R_3 + \gamma^2 R_4 + \gamma^3 R_5 = 2 + 0.5 \cdot 4 + 0.5^2 \cdot 3 + 0.5^3 \cdot 2$$

$$G_1 = 2 + 2 + 0.75 + 0.25 = 5$$

G0=



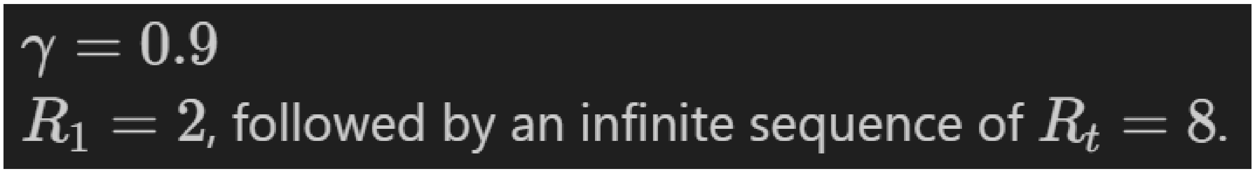
$$G_0 = R_1 + \gamma R_2 + \gamma^2 R_3 + \gamma^3 R_4 + \gamma^4 R_5$$

$$G_0 = -1 + 0.5 \cdot 2 + 0.5^2 \cdot 4 + 0.5^3 \cdot 3 + 0.5^4 \cdot 2$$

$$G_0 = -1 + 1 + 1 + 0.375 + 0.125 = 1.5$$

b. The same formula is used for the infinite sequence

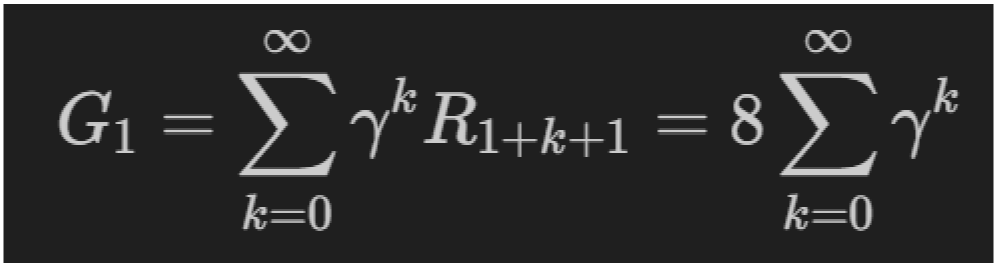
Given:



$\gamma = 0.9$

$R_1 = 2$, followed by an infinite sequence of $R_t = 8$.

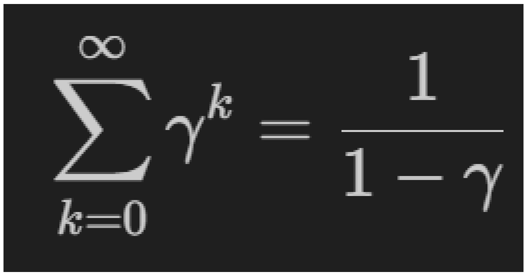
G1=



$$G_1 = \sum_{k=0}^{\infty} \gamma^k R_{1+k+1} = 8 \sum_{k=0}^{\infty} \gamma^k$$

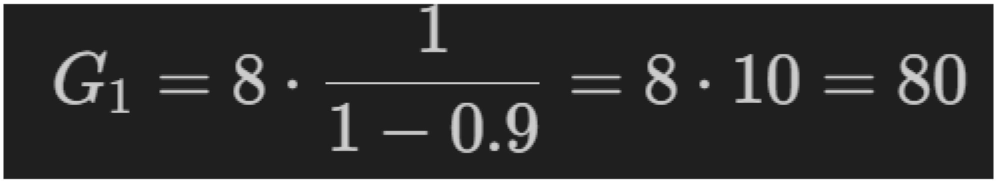
Since R2 onwards is 8, we can take the 8 outside the summation

Using the geometric series formula:



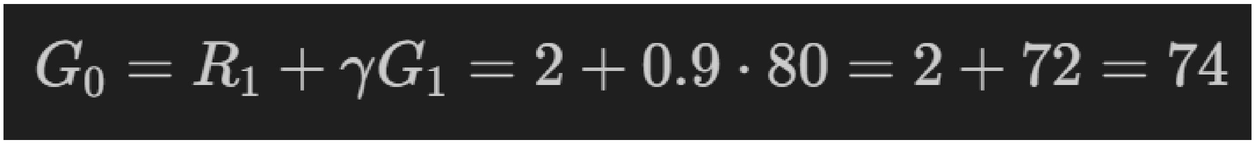
$$\sum_{k=0}^{\infty} \gamma^k = \frac{1}{1-\gamma}$$

Therefore,



$$G_1 = 8 \cdot \frac{1}{1-0.9} = 8 \cdot 10 = 80$$

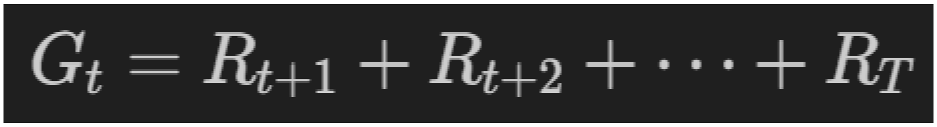
And G0=



$$G_0 = R_1 + \gamma G_1 = 2 + 0.9 \cdot 80 = 2 + 72 = 74$$

Q3 – The RL Objective

Based on the equation 3.7



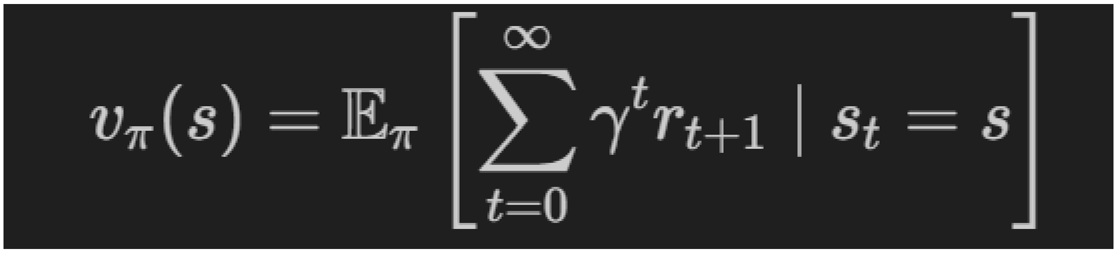
$$G_t = R_{t+1} + R_{t+2} + \cdots + R_T$$

The return, irrespective of what path the agent takes would be 1. $R_T = 1$ is the only term in that equation. This is the sparse rewards case, thus the agent does not really have the information it needs to get the optimal path. If we either introduce a -1 reward for states that are not the goal states, or implement a discounted return for all states that are not the goal state. The agent will have incentive and information to find the optimal path for escaping from the maze. Thus, the issue in this case is simply that there is no reward data to optimize the path. If we introduce something like that, the optimal path will be found.

Q4 – Modifying the reward function

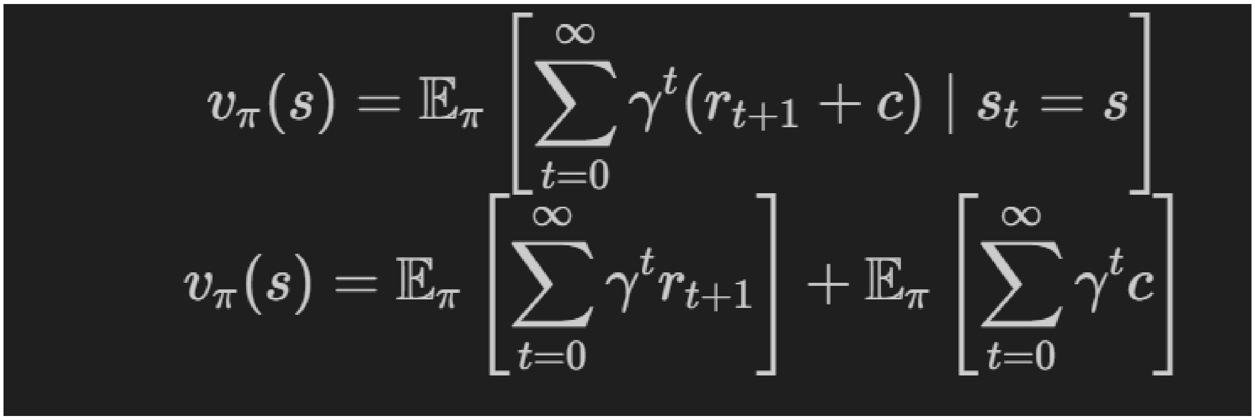
The signs of the reward are not important. Because the rewards play a role in decision making, as long as undesirable states have a smaller reward than desirable states, the agent will learn to maximize its return by reaching the desirable (or goal) states. Thus the intervals between the rewards is important.

Consider the bellman equation for the value function



$$v_\pi(s) = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t r_{t+1} \mid s_t = s\right]$$

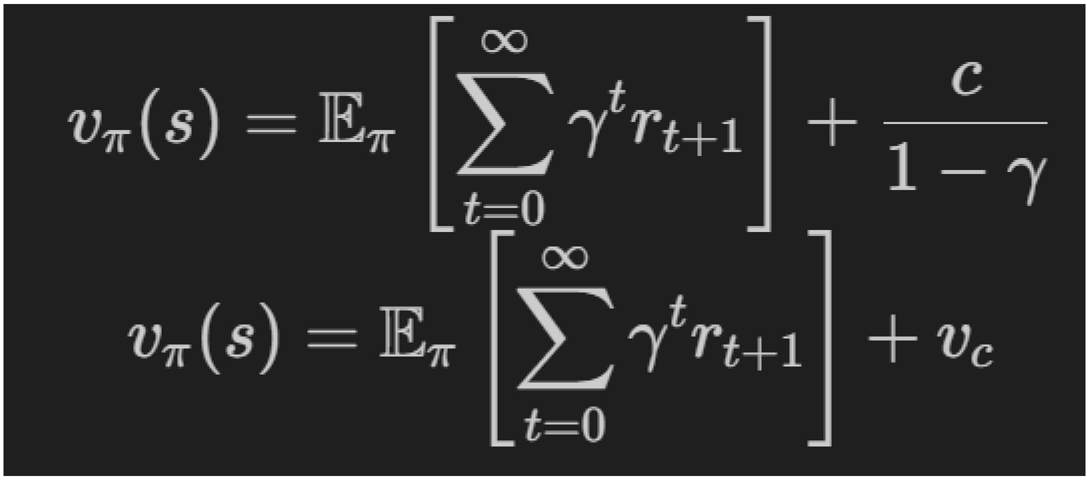
Adding constant c to the rewards



$$v_\pi(s) = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t (r_{t+1} + c) \mid s_t = s\right]$$

$$v_\pi(s) = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t r_{t+1}\right] + \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t c\right]$$

Solving the second term using infinite geometric series

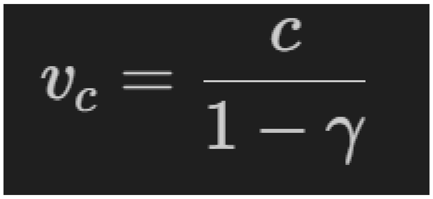


$$v_\pi(s) = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty}\gamma^t r_{t+1}\right] + \frac{c}{1-\gamma}$$

$$v_\pi(s) = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty}\gamma^t r_{t+1}\right] + v_c$$

Notice that the second term is a constant. Thus, adding a constant offset to the rewards just shifts the value function by a constant v_c.

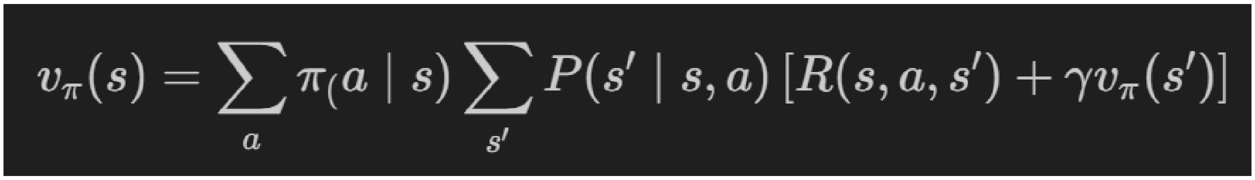
The value of v_c is:



$$v_c = \frac{c}{1-\gamma}$$

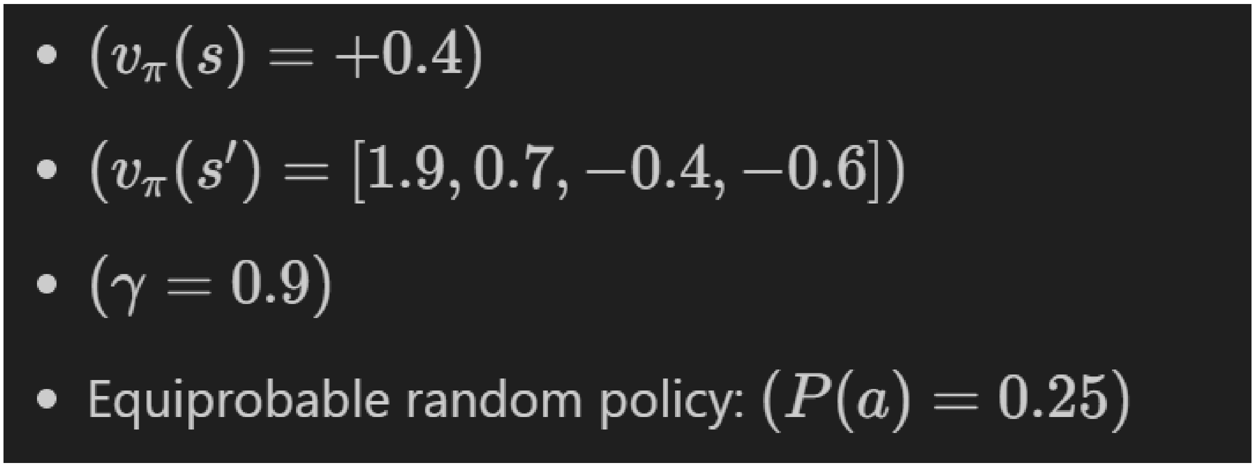
Q5 – Bellman equation

a. Using the following Bellman equation



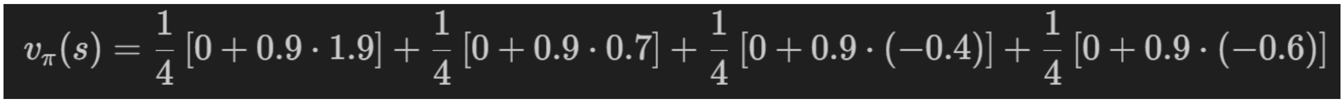
$$v_\pi(s) = \sum_a \pi_{(}a \mid s) \sum_{s'} P(s' \mid s, a)\left[R(s, a, s') + \gamma v_\pi(s')\right]$$

Given that

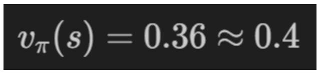


- $(v_\pi(s) = +0.4)$
- $(v_\pi(s') = [1.9, 0.7, -0.4, -0.6])$
- $(\gamma = 0.9)$
- Equiprobable random policy: $(P(a) = 0.25)$

Substituting in the equation



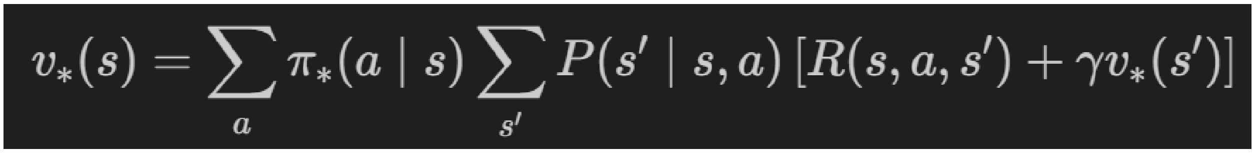
$$v_\pi(s) = \frac{1}{4}[0 + 0.9 \cdot 1.9] + \frac{1}{4}[0 + 0.9 \cdot 0.7] + \frac{1}{4}[0 + 0.9 \cdot (-0.4)] + \frac{1}{4}[0 + 0.9 \cdot (-0.6)]$$



$$v_\pi(s) = 0.36 \approx 0.4$$

Thus, the result is verified for approximation to one decimal point.

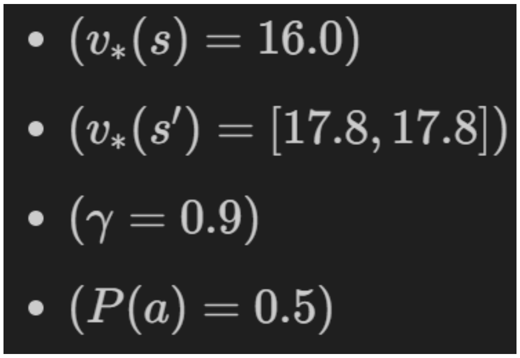
b. For the optimal policy,



$$v_*(s) = \sum_a \pi_*(a \mid s) \sum_{s'} P(s' \mid s, a)\left[R(s, a, s') + \gamma v_*(s')\right]$$

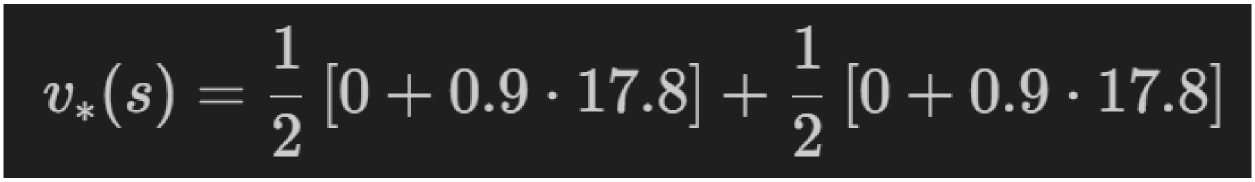
According to the figure, the state we are looking at has equiprobable chance to go in two directions, LEFT and UP.

Thus,

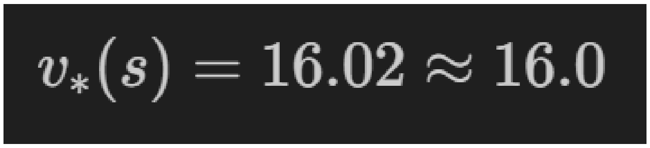


- $(v_*(s) = 16.0)$
- $(v_*(s') = [17.8, 17.8])$
- $(\gamma = 0.9)$
- $(P(a) = 0.5)$

Substituting in the equation



$$v_*(s) = \frac{1}{2}\left[0 + 0.9 \cdot 17.8\right] + \frac{1}{2}\left[0 + 0.9 \cdot 17.8\right]$$



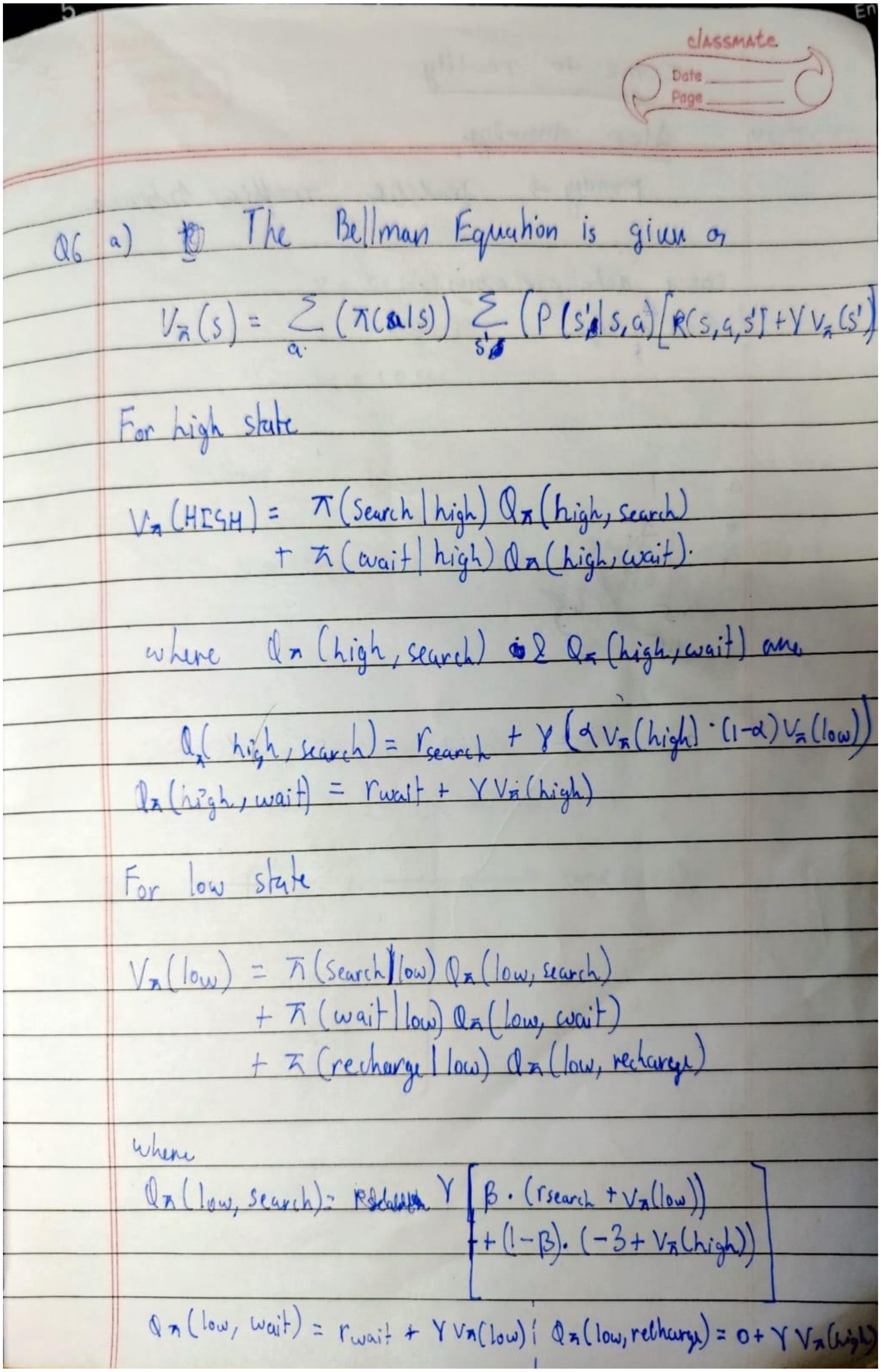
$$v_*(s) = 16.02 \approx 16.0$$

Thus, the result is verified for approximation to one decimal point.

Adnan Amir

Q6 – Solving for the value function

a.



Q6 a) The Bellman Equation is given as

$$V_\pi(s) = \sum_{a} (\pi(a|s)) \sum_{s'} (P(s'|s,a)[R(s,a,s'] + \gamma V_\pi(s')$$

For high state

$$V_\pi(HIGH) = \pi(Search|high)\, Q_\pi(high, Search) + \pi(wait|high)\, Q_\pi(high, wait).$$

where $Q_\pi(high, search)$ & $Q_\pi(high, wait)$ are

$$Q_\pi(high, search) = r_{search} + \gamma(\alpha V_\pi(high] \cdot (1-\alpha) V_\pi(low))$$

$$Q_\pi(high, wait) = r_{wait} + \gamma V_\pi(high)$$

For low state

$$V_\pi(low) = \pi(Search|low)\, Q_\pi(low, search) + \pi(wait|low)\, Q_\pi(low, wait) + \pi(recharge|low)\, Q_\pi(low, recharge)$$

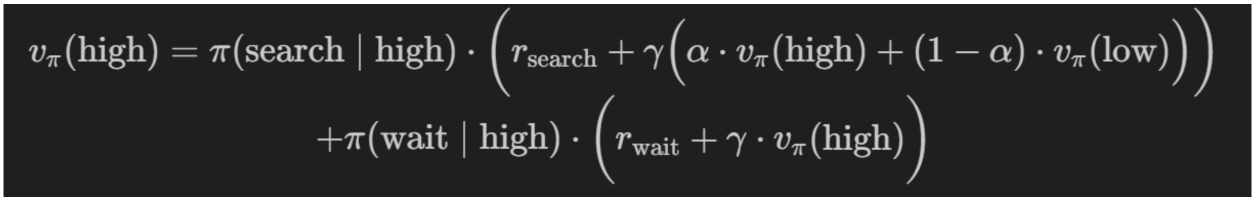
where

$$Q_\pi(low, Search) = \gamma \left[\beta \cdot (r_{search} + V_\pi(low)) + (1-\beta) \cdot (-3 + V_\pi(high)) \right]$$

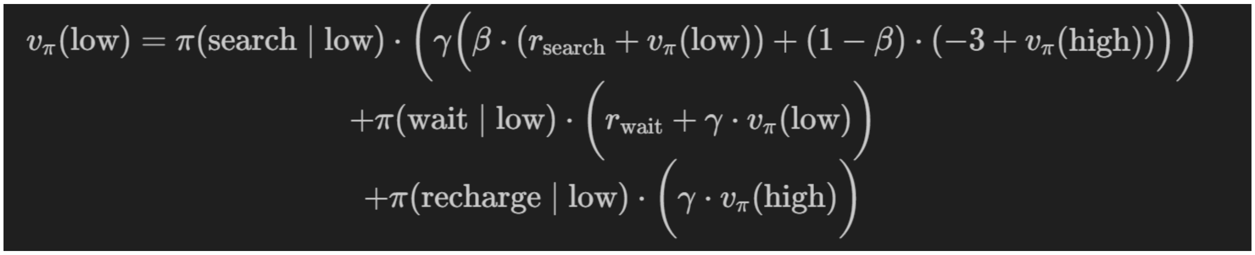
$Q_\pi(low, wait) = r_{wait} + \gamma V_\pi(low)$; $Q_\pi(low, recharge) = 0 + \gamma V_\pi(high)$

Adnan Amir

Attaching the final equations using latex, incase my handwriting is not legible

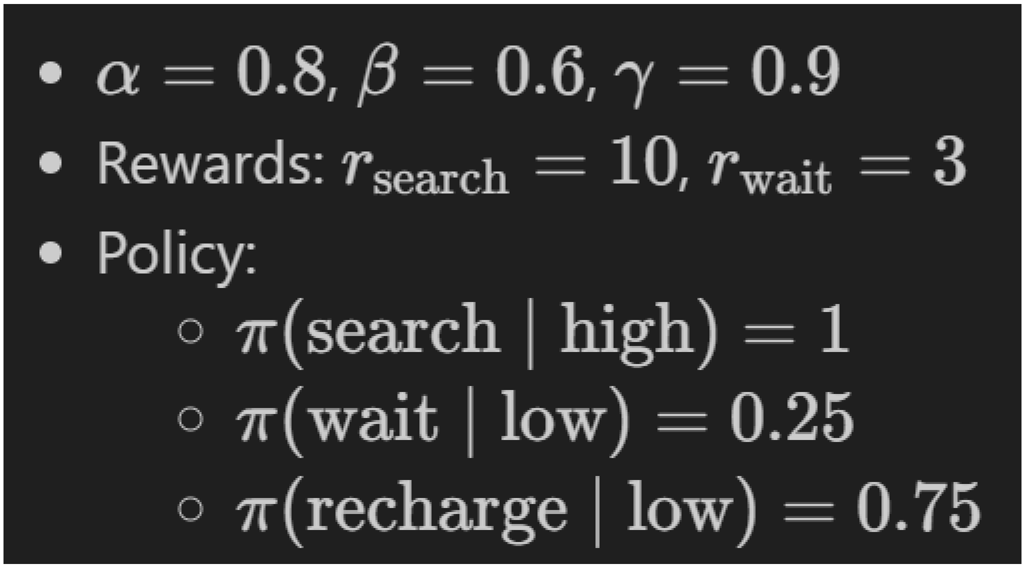


$$\begin{aligned} v_\pi(\text{high}) = \pi(\text{search} \mid \text{high}) \cdot \Big(r_{\text{search}} + \gamma\big(\alpha \cdot v_\pi(\text{high}) + (1-\alpha) \cdot v_\pi(\text{low})\big)\Big) \\ +\pi(\text{wait} \mid \text{high}) \cdot \Big(r_{\text{wait}} + \gamma \cdot v_\pi(\text{high})\Big) \end{aligned}$$



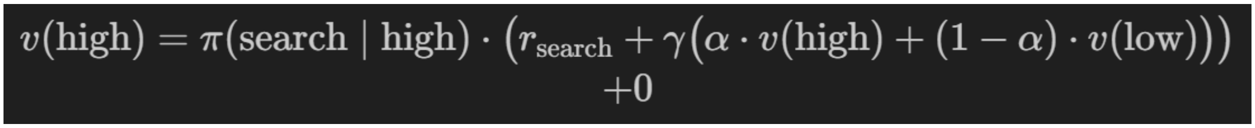
$$\begin{aligned} v_\pi(\text{low}) = \pi(\text{search} \mid \text{low}) \cdot \Big(\gamma\big(\beta \cdot (r_{\text{search}} + v_\pi(\text{low})) + (1-\beta) \cdot (-3 + v_\pi(\text{high}))\big)\Big) \\ +\pi(\text{wait} \mid \text{low}) \cdot \Big(r_{\text{wait}} + \gamma \cdot v_\pi(\text{low})\Big) \\ +\pi(\text{recharge} \mid \text{low}) \cdot \Big(\gamma \cdot v_\pi(\text{high})\Big) \end{aligned}$$

b. Given that

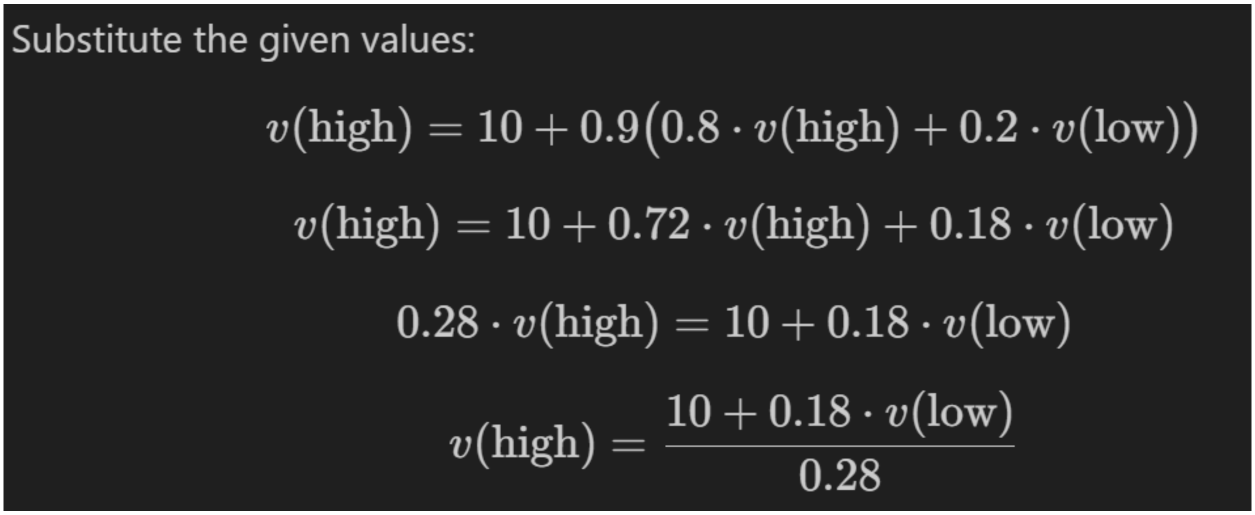


- $\alpha = 0.8$, $\beta = 0.6$, $\gamma = 0.9$
- Rewards: $r_{\text{search}} = 10$, $r_{\text{wait}} = 3$
- Policy:
  - $\pi(\text{search} \mid \text{high}) = 1$
  - $\pi(\text{wait} \mid \text{low}) = 0.25$
  - $\pi(\text{recharge} \mid \text{low}) = 0.75$

For high state, policy always searches. Therefore,



$$\begin{aligned} v(\text{high}) = \pi(\text{search} \mid \text{high}) \cdot \big(r_{\text{search}} + \gamma(\alpha \cdot v(\text{high}) + (1-\alpha) \cdot v(\text{low}))\big) \\ +0 \end{aligned}$$



Substitute the given values:

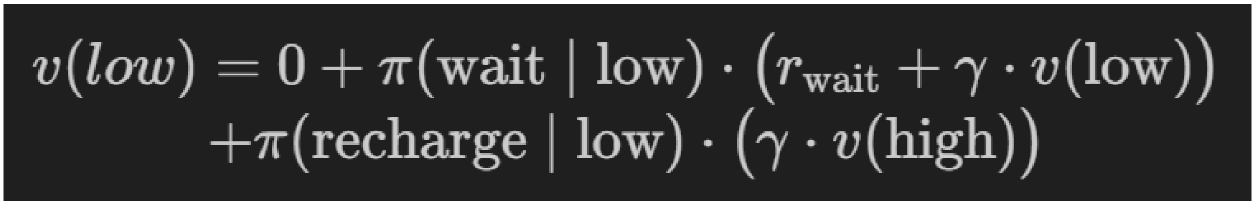
$$v(\text{high}) = 10 + 0.9\big(0.8 \cdot v(\text{high}) + 0.2 \cdot v(\text{low})\big)$$

$$v(\text{high}) = 10 + 0.72 \cdot v(\text{high}) + 0.18 \cdot v(\text{low})$$

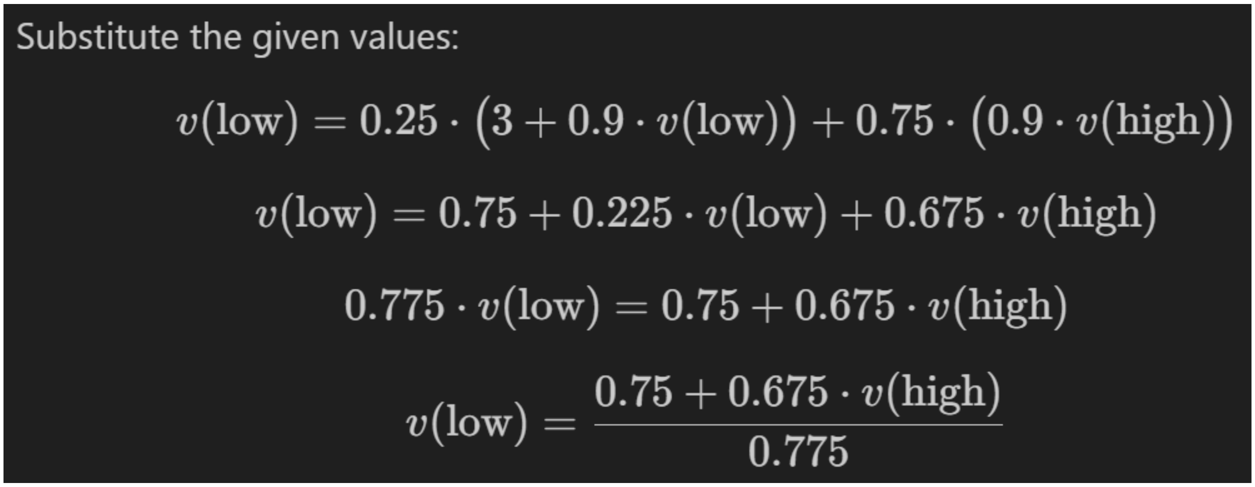
$$0.28 \cdot v(\text{high}) = 10 + 0.18 \cdot v(\text{low})$$

$$v(\text{high}) = \frac{10 + 0.18 \cdot v(\text{low})}{0.28}$$

Similarly, for low state, The policy only waits or recharges. Therefore equation becomes



$$v(low) = 0 + \pi(\text{wait} \mid \text{low}) \cdot \big(r_{\text{wait}} + \gamma \cdot v(\text{low})\big) \\ +\pi(\text{recharge} \mid \text{low}) \cdot \big(\gamma \cdot v(\text{high})\big)$$



Substitute the given values:

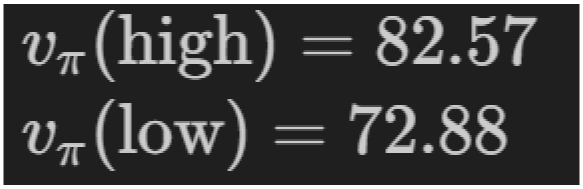
$$v(\text{low}) = 0.25 \cdot \big(3 + 0.9 \cdot v(\text{low})\big) + 0.75 \cdot \big(0.9 \cdot v(\text{high})\big)$$

$$v(\text{low}) = 0.75 + 0.225 \cdot v(\text{low}) + 0.675 \cdot v(\text{high})$$

$$0.775 \cdot v(\text{low}) = 0.75 + 0.675 \cdot v(\text{high})$$

$$v(\text{low}) = \frac{0.75 + 0.675 \cdot v(\text{high})}{0.775}$$

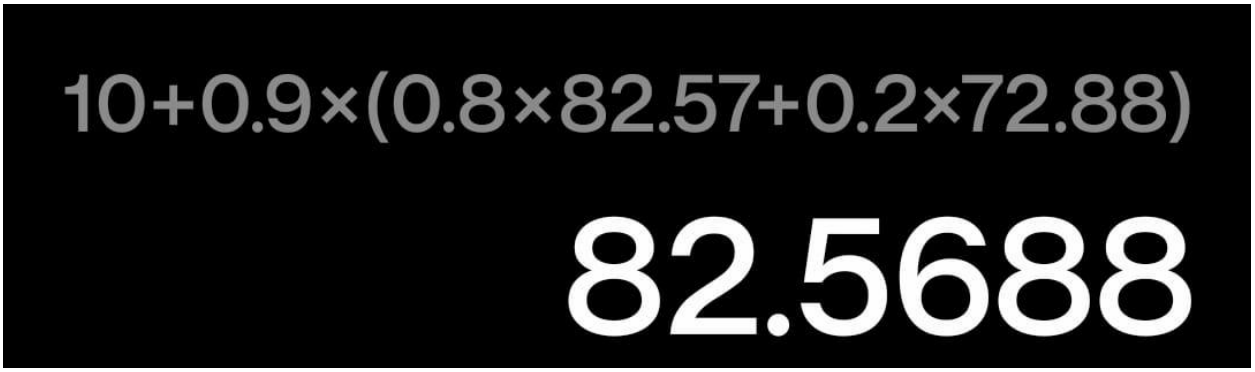
Solving the two equations simultaneously we get:



$$v_\pi(\text{high}) = 82.57$$
$$v_\pi(\text{low}) = 72.88$$

Substituting in step 1 of both bellman equations, The equations are satisfied.

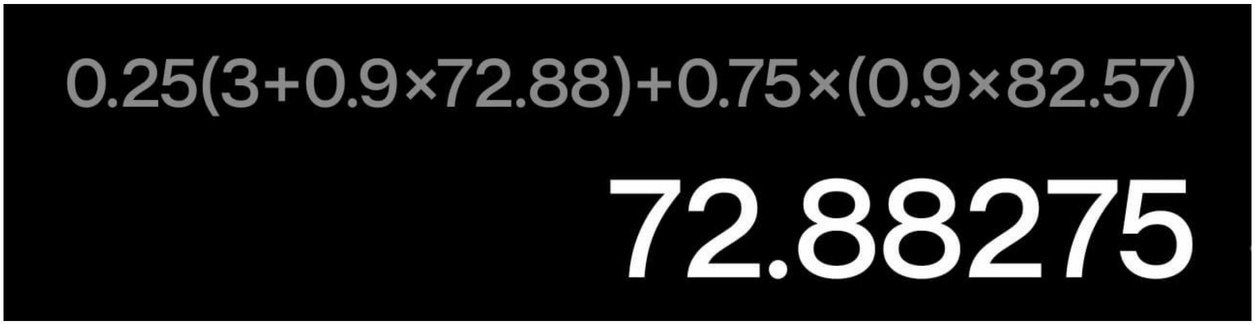
For the v(high)



10+0.9×(0.8×82.57+0.2×72.88)

82.5688

For the v(low)



0.25(3+0.9×72.88)+0.75×(0.9×82.57)

72.88275