Q0: (a) Yes. Either anonymized or not is okay

(b) Same for EX0 and EX1

Q1:

(a)

$$V_*(s) = \max_a q_*(s,a)$$

(b) $q_*(a,s) = \sum_{s',r} p(s',r|s,a)(r + \gamma V_*(s'))$

(c) $\pi_*(s) = \arg\max_a q_*(s,a)$

(d) $\pi_*(s) = \arg\max_a \sum_{s',r} p(s',r|s,a)(r + \gamma V_*(s'))$

(e) $V_\pi(s) = \sum_a \pi(a|s) [\sum_{s',r} p(s',r|s,a) \cdot r + \sum_{s',r} p(s',r|s,a) \cdot \gamma \cdot V_\pi(s')]$

$= \sum_a \pi(a|s)(r(s,a) + \sum_{s'} p(s'|s,a) \cdot \gamma V_\pi(s'))$

$V_*(s) = \max_a (r(s,a) + \sum_{s'} p(s'|s,a) \cdot \gamma V_*(s'))$

$q_\pi(s,a) = r(s,a) + \sum_{s'} p(s'|s,a) \cdot \gamma \sum_{a'} \pi(a'|s) \cdot q(s',a')$

$q_*(s,a) = r(s,a) + \sum_{s'} p(s'|s,a) \gamma \max_{a'} q_*(s',a')$

Q2:

(a). The policy keeps switching between two optimal ones because the original pseudo code use actions as criteria to judge stability of the policy, while actually we should stop the iteration when we know we already have an optimal one

fix: change the second line from the last to:

if $\sum_{s',r} p(s',r|s,old\_action)(r + \gamma V(s')) \neq \sum_{s',r} p(s',r|s,\pi(s))(r + \gamma V(s'))$, policy-stable $\leftarrow$ false

(b). There isn't, because value iteration derives the optimal V^* first, and then calculate the policy, so there isn't switching action.

Q3:

(a). 1. Initialization

$Q(s,a) \in \mathbb{R}$ and $\pi(s) \in A(s)$ arbitarily for all $s \in S$

2. policy evaluation

Loop:

$\Delta \leftarrow 0$

Loop for each $s \in S$

$a \leftarrow \pi(s)$

$q \leftarrow Q(s,a)$

$Q(s,a) \leftarrow \sum_{s',r} p(s',r|s,a)[r + \gamma \sum_{a'} \pi(a'|s')(Q(s',a')]$

$\Delta = \max(\Delta, |q - Q(s,a)|)$

until $\Delta < \theta$ (a small value for accuracy)

3. policy improvement

policy_stable $\leftarrow$ true

For each $s \in S$

old_action $\leftarrow \pi(s)$

$\pi(s) \leftarrow \arg\max_a q(s,a)$

If $\sum_{s',r} p(s',r|s,\pi(s))[r + \gamma \sum_{a'} \pi(a'|s') q(a',s')] \neq \sum_{s',r} p(s',r|s,\text{old\_action})[r + \gamma \sum_{a'} \pi(a'|s') q(a',s')]$,

then policy_stable $\leftarrow$ false

If policy_stable, then stop and return $Q \doteq q_*$, and $\pi \doteq \pi_*$; else go to 2.

(b) init: threshold $\theta > 0$; $Q(s,a)$, for all $s \in S^+$, arbitarily $Q(s,a) = 0$ when s is terminal.

Loop:

$\Delta \leftarrow 0$

Loop for each $s \in S$:

$a \leftarrow \arg\max_a q(s,a)$

$q \leftarrow Q(s,a)$

$Q(s,a) \leftarrow \sum_{s',r} p(s',r|s,a)[r + \gamma \max_{a'} q(s',a')]$

$\Delta \leftarrow \max(\Delta, |q - Q(s,a)|)$

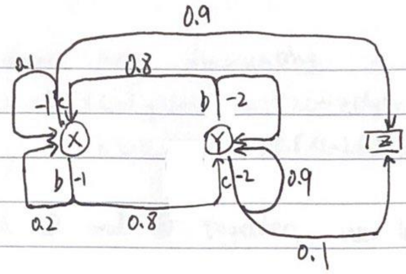
until $\Delta < \theta$.

Output deterministic policy, $\pi \approx \pi_*$, such that

$\pi(s) = \arg\max_a \sum_{s',r} p(s',r|s,a)[r + \gamma\, q(s,\pi(s'))]$

Q4:

(a)



Since no matter what action we choose at Y, it will always be worse than in X, so qualitatively optimal policy for X should be **C** because by doing this it can avoid entering Y, which is worse. For Y, it's not very clear to me, since whether entering X or C it's always a improvement, and I think whether b or c is optimal for y is depended on the discounting factor.

(b) $V_\pi(s) = \sum_a \pi(a|s) \cdot \sum_{s',r} p(s',r|s,a)(r + V_\pi(s'))$

$$\begin{cases} V_\pi(x) = 0.1\times(-1) + 0.9\times(-1+V_\pi(x)) \\ V_\pi(y) = 0.1\times(-2) + 0.9\times(-2+V_\pi(y)) \end{cases} \Rightarrow \begin{cases} V_\pi(x) = -10 \\ V_\pi(y) = -20 \end{cases}$$ evaluation 1

$\pi(s) = \arg\max_a \sum_{s',r} p(s',r|s,a)(r + \gamma V_\pi(s'))$

$V_{\pi b}(x) = 0.2(-1+V_\pi(x)) + 0.8(-1+V_\pi(y)) = -1-2-16 = -19$

$V_{\pi b}(y) = 0.2(-2+V_\pi(y)) + 0.8(-2+V_\pi(y)) = -2-4-8 = -14$.

$\because \begin{cases} V_{\pi c}(x) = -10 \\ V_{\pi c}(y) = -20 \end{cases}$

$\therefore \pi'(x) = c, \pi'(y) = b$. improvement 1

$$\begin{cases} V_\pi(x) = 0.1\times(-1) + 0.9\times(-1+V_\pi(x)) \\ V_\pi(y) = 0.2\times(-2+V_\pi(y)) + 0.8\times(-2+V_\pi(x)) \end{cases} \Rightarrow \begin{cases} V_\pi(x) = -10 \\ V_\pi(y) = -12.5 \end{cases}$$ evaluation 2.

$V_{\pi b}(x) = 0.2\times(-1+V_\pi(x)) + 0.8(-1+V_\pi(y)) = -13$

$V_{\pi c}(y) = 0.1\times(-2) + 0.9\times(-2+V_\pi(y)) = -13.25$

$\because \begin{cases} V_{\pi c}(x) = -10 \\ V_{\pi b}(y) = -12.5 \end{cases}$ improvement 2.

$\therefore \pi''(x) = c = \pi'(x), \pi''(y) = b = \pi'(y)$

$\therefore$ stablized, $\pi_*(x) = c$, $\pi_*(y) = b$.

(C) when there isn't discounting

$$\begin{cases} V_\pi(x) = 0.2\times(-1+V_\pi(x)) + 0.8\times(-1+V_\pi(y)) & ① \\ V_\pi(y) = 0.2\times(-2+V_\pi(y)) + 0.8\times(-2+V_\pi(x)) & ② \end{cases}$$

add ① and ② together, we have $0=-3$, which is obviously wrong.

After adding discounting factor γ!

$$\begin{cases} V_\pi(x) = 0.2\times(-1+\gamma V_\pi(x)) + 0.8\times(-1+\gamma V_\pi(y)) \\ V_\pi(y) = 0.2\times(-2+\gamma V_\pi(y)) + 0.8\times(-2+\gamma V_\pi(x)) \end{cases}$$

$$\Rightarrow \begin{cases} V_\pi(y) = \dfrac{-2-0.4\gamma}{1-0.6\gamma^2-0.4\gamma} & = V_{\pi b}(y) \\ V_\pi(x) = \dfrac{-1+0.28\gamma^2-1.2\gamma}{(1-0.6\gamma^2-0.4\gamma)(1-0.2\gamma)} & = V_{\pi b}(x) \end{cases}$$

$$\begin{cases} V_{\pi c}(x) = 0.1\times(-1) + 0.9\times(-1+V_\pi(x)) \\ \quad = -1+0.9V_\pi(x) & \rightarrow \text{ if } V_\pi(x) < -10, \text{ then } \pi(x) = C \\ V_{\pi c}(y) = -2+0.9V_\pi(y) & \rightarrow \text{ if } V_\pi(y) < -20, \text{ then } \pi(y) = C. \end{cases}$$

By observing $V_{\pi b}(y)$, we can find that $(-\infty, -2]$ is part of it value range, because when $\gamma \to 1$, $V_\pi(y) \to -\frac{2}{0}$, and when $\gamma=0$, $V_\pi(y) = -2$, and it's continuous on $\gamma \in (0,1)$, so we can definitely control the value of γ to make it higher or lower than -20, and thus control the optimal policy for y. Similarly, we can do the same thing on x.

In conclution, optimal policy is depended on the discount factor.

Q5:

(a)

```
[[ 3.31  8.79  4.43  5.32  1.49]
 [ 1.52  2.99  2.25  1.91  0.55]
 [ 0.05  0.74  0.67  0.36 -0.4 ]
 [-0.97 -0.43 -0.35 -0.58 -1.18]
 [-1.86 -1.34 -1.23 -1.42 -1.97]]
```

Same as Figure 3.2

(b)

```
[[21.98 24.42 21.98 19.42 17.48]
 [19.78 21.98 19.78 17.8  16.02]
 [17.8  19.78 17.8  16.02 14.42]
 [16.02 17.8  16.02 14.42 12.98]
 [14.42 16.02 14.42 12.98 11.68]]
{'[0, 0]': ['r'], '[0, 1]': ['u', 'd', 'l', 'r'], '[0, 2]': ['l'],
'[0, 3]': ['u', 'd', 'l', 'r'], '[0, 4]': ['l'], '[1, 0]': ['u',
'r'], '[1, 1]': ['u'], '[1, 2]': ['u', 'l'], '[1, 3]': ['l'], '[1,
4]': ['l'], '[2, 0]': ['u', 'r'], '[2, 1]': ['u'], '[2, 2]': ['u',
'l'], '[2, 3]': ['u', 'l'], '[2, 4]': ['u', 'l'], '[3, 0]': ['u',
'r'], '[3, 1]': ['u'], '[3, 2]': ['u', 'l'], '[3, 3]': ['u', 'l'],
'[3, 4]': ['u', 'l'], '[4, 0]': ['u', 'r'], '[4, 1]': ['u'], '[4,
2]': ['u', 'l'], '[4, 3]': ['u', 'l'], '[4, 4]': ['u', 'l']}
```

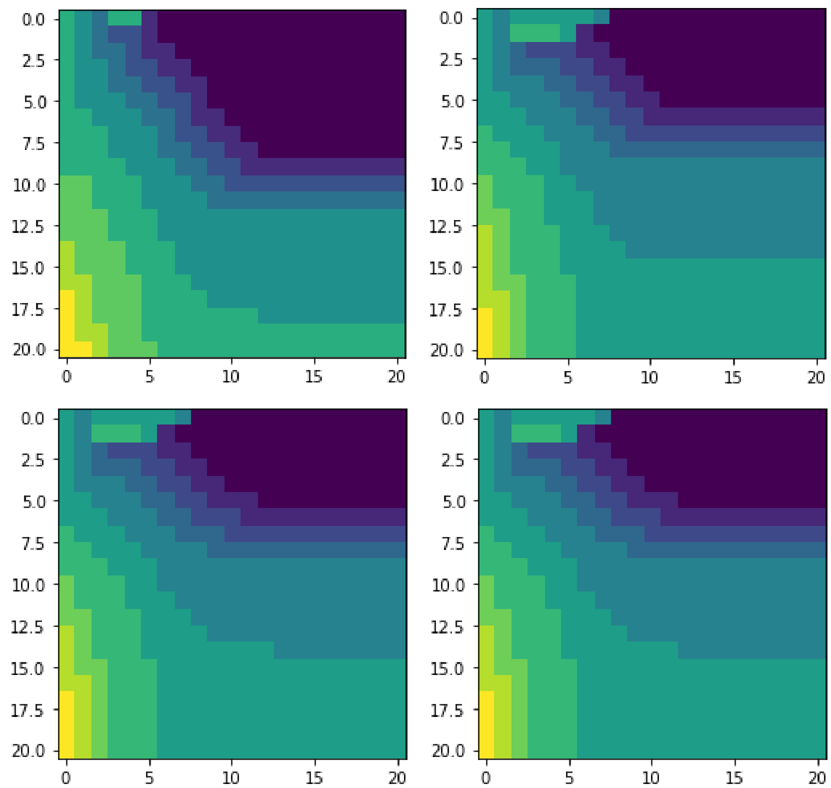
Same as Figure 3.5. The dictionary contains derived optimal policy, which key is state and value is action.

(c)

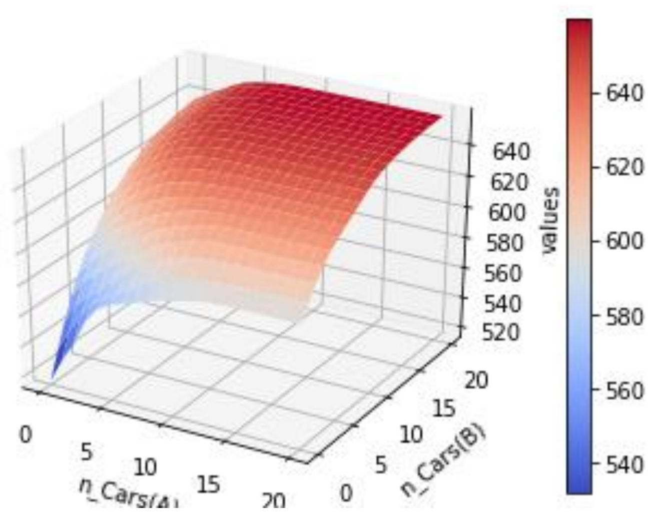
```
[[21.98 24.42 21.98 19.42 17.48]
 [19.78 21.98 19.78 17.8  16.02]
 [17.8  19.78 17.8  16.02 14.42]
 [16.02 17.8  16.02 14.42 12.98]
 [14.42 16.02 14.42 12.98 11.68]]
{'[0, 0]': ['r'], '[0, 1]': ['u', 'd', 'l', 'r'], '[0, 2]': ['l'],
'[0, 3]': ['u', 'd', 'l', 'r'], '[0, 4]': ['l'], '[1, 0]': ['u',
'r'], '[1, 1]': ['u'], '[1, 2]': ['u', 'l'], '[1, 3]': ['l'], '[1,
4]': ['l'], '[2, 0]': ['u', 'r'], '[2, 1]': ['u'], '[2, 2]': ['u',
'l'], '[2, 3]': ['u', 'l'], '[2, 4]': ['u', 'l'], '[3, 0]': ['u',
'r'], '[3, 1]': ['u'], '[3, 2]': ['u', 'l'], '[3, 3]': ['u', 'l'],
'[3, 4]': ['u', 'l'], '[4, 0]': ['u', 'r'], '[4, 1]': ['u'], '[4,
2]': ['u', 'l'], '[4, 3]': ['u', 'l'], '[4, 4]': ['u', 'l']}
```

Q6:

(a) Yellow indicates positive, blue indicates negative. The color in the middle of these four graphs indicates ‘0’.



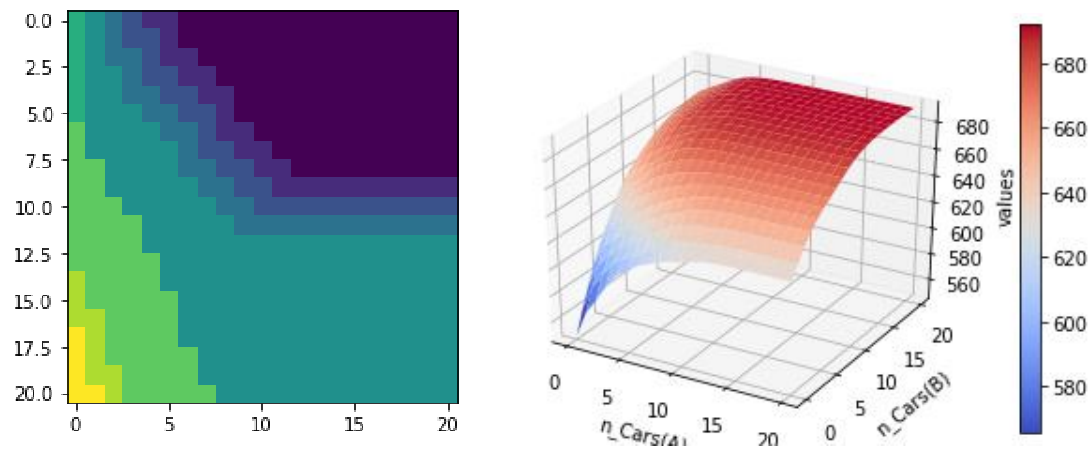
π_1 to π_4 (top-left to bottom-right), with initial π_0 has 0 as action for every state



3D-plot of v_{π_4}, maximum state value located at (20,20)

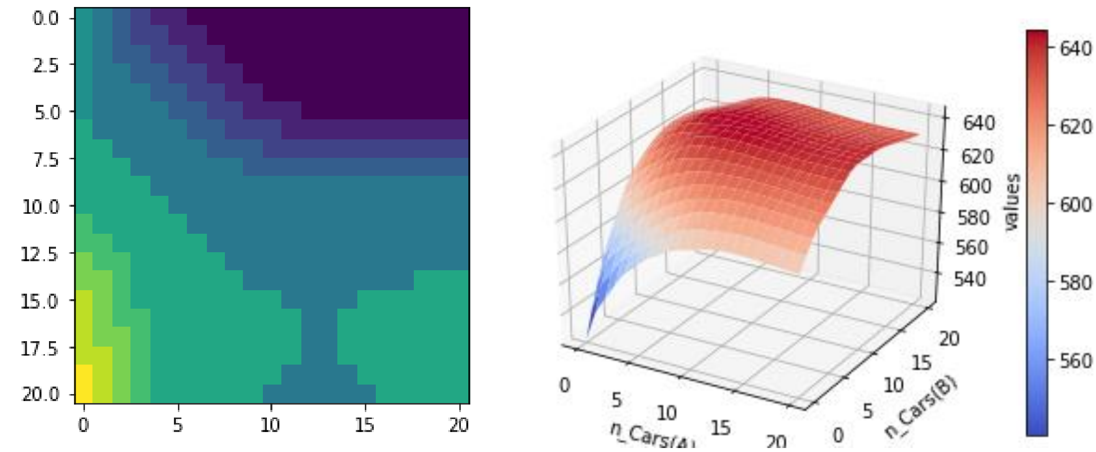
(b) The reward will change when making those modifications. When an employee is able to help switching cars the cost on switching in one of the directions will decrease by one, so I assume that employee lives at the second place and thus changed the reward when moves cars from the first place to the second from **reward = n_rented*10 – 2*num_movedCars** to **reward = n_rented*10 – 2*max((num_movedCars-1),0)**, and reward function for the other direction remains unchanged. In this case the optimal policy should be tend to switch more cars in one direction.

For the second case, when the parking lot charge for extra fees there will be an extra cost which result in lower reward, so I subtracted an extra term parking_fee if the total number of cars in that place is more than 10, as **n_rented*10 – 2*num_movedCars – parking_fee**. And I think the optimal policy should tend to keep less cars in both locations.



π_4 and 3D-plot of v_{π_4} of employee problem

As a result, the policy tends to move more cars in one direction and the other direction remains the same. The state values are also rise in general, since the cost of moving cars decreased, while the peak value still appears in (20,20).



π_4 and 3D-plot of v_{π_4} of parking fee problem

As a result, the policy tends to keep less cars in both locations, and sometimes it doesn't even move the cars when one of the locations has more than 15 cars because it wants to prevent being charged for extra parking fee for the other parking lot. The state values are also decreased in general, since the reward has been subtracted by extra parking fee, and it also impact the peak value which appears to be (10,10). It makes sense because keeping 10 cars in both parking lot is the best state that maximizing number of cars and avoiding extra parking fee.

Q7.

(a). if $\max_a f(a) - \max_a g(a) \geq 0$:

$$|\max_a f(a) - \max_a g(a)| = \max_a f(a) - \max_a g(a)$$

$\because \max_a g(a) \geq g(x)$ for all x

$\therefore \max_a f(a) - \max_a g(a) \leq \max_a f(a) - g(x)$, for all x,

$\therefore \max_a f(a) - \max_a g(a) \leq \max_a f(a) - g(a)$ ①

Say, $a_1 = \arg\max f(a)$, $a_2 = \arg\max (f(a) - g(a))$

then $f(a_1) - g(a_1) \leq f(a_2) - g(a_2)$ ② could be easily proved.

② ⇒ ①: where $f(a_1) - g(a_1) = \max_a f(a) - g(a)$,

$$f(a_2) - g(a_2) = \max_a (f(a) - g(a))$$

$$\Rightarrow \max_a f(a) - g(a) \leq \max_a (f(a) - g(a))$$

$\therefore$ proved.

When $\max_a f(a) - \max_a g(a) < 0$

substitute $f(a) \leftarrow g(a)$, and $g(a) \leftarrow f(a)$, and we can prove it in the same way.

$$\therefore \quad |\max_a f(a) - \max_a g(a)| \leq \max_a |f(a) - g(a)|.$$

(b). $$|\beta V_i(s) - \beta V_i'(s)| = \left|\max_a \sum_{s',r} p(s',r|s,a)[r + V_i(s')] - \max_a \sum_{s',r} p(s',r|s,a)[r + V_i'(s')]\right| \quad ①$$

use equation proved from (a)

$$① \leq \max_a \left|\sum_{s',r} p(s',r|s,a)[r + \gamma V_k(s')] - \sum_{s',r} p(s',r|s,a)[r + \gamma V_k'(s')]\right|$$

$$= \max_a \left|\sum_{s',r} p(s',r|s,a)(r + \gamma V_k(s') - r - \gamma V_k'(s')\right|$$

$$= \max_a \left|\sum_{s',r} p(s',r|s,a) \cdot \gamma (V_k(s') - V_k'(s'))\right| \quad \text{hold for all } s \in S$$

for n-length Vector V_i and V_i':

$$\|\beta V_i - \beta V_i'\| = \max \begin{Bmatrix} |\beta V_i(s_1) - \beta V_i'(s_1)| \\ |\beta V_i(s_2) - \beta V_i'(s_2)| \\ \vdots \\ |\beta V_i(s_n) - \beta V_i'(s_n)| \end{Bmatrix} \leq \max \begin{Bmatrix} \max_a \left|\sum_{s',r} p(s',r|s,a)\gamma(V_i(s_1) - V_i'(s_1))\right| \\ \max_a \left|\sum_{s',r} p(s',r|s,a)\gamma(V_i(s_2) - V_i'(s_2))\right| \\ \vdots \\ \max_a \left|\sum_{s',r} p(s',r|s,a)\gamma(V_i(s_n) - V_i'(s_n))\right| \end{Bmatrix} \quad ②$$

$$② \le \max \begin{Bmatrix} \max_a \left| \sum_{s',r} \cdot p(s',r|s,a) \cdot \gamma \|V_i - V_i'\| \right. \\ \text{duplicate.} \quad \vdots \quad \times n. \\ \vdots \\ \max_a \left| \sum_{s',r} \cdot p(s',r|s,a) \cdot \gamma \|V_i - V_i'\| \right. \end{Bmatrix}$$

$$= \max. \begin{Bmatrix} \gamma \|V_i - V_i'\| \\ \gamma \|V_i - V_i'\| \\ \vdots \times n \\ \gamma \|V_i - V_i'\| \end{Bmatrix} = \gamma \| V_K - V_K' \|.$$

$\therefore$ proved.

(C) Banach fixed point therom: $d(T(x), T(y)) \le q\, d(x,y)$

in our case: $d(x,y) = \|x - y\|$,

$T(x) = \beta x$, β is contracting mapping

$q = \gamma$

use equation from (b): $\|\beta V_i - \beta V_i'\|_\infty \le \gamma \|V_i - V_i'\|_\infty$ ③

proof:

first for any positive integer n; $\|V_{n+1} - V_n\| \le \gamma^n \|V_1 - V_0\|$ (use ③).

then, for any m, n:

$$\|V_m - V_n\| \le \|V_m - V_{m-1}\| + \|V_{m-1} - V_{m-2}\| + \cdots + \|V_{n+1}, V_n\|$$

$$\le \gamma^{m-1} \|V_1 - V_0\| + \gamma^{m-2} \|V_1 - V_0\| + \cdots \gamma^n \|V_1, V_0\|$$

$$= \gamma^n \|V_1 - V_0\| \sum_{k=0}^{m-n-1} \gamma^k$$

$$\le \gamma^n \|V_1 - V_0\| \sum_{k=0}^{\infty} \gamma^k.$$

$$= \gamma^n \|V_1 - V_0\| \frac{1}{1-\gamma}$$

We can find a large N that $q^N < \frac{\varepsilon(1-q)}{\|V_1 - V_0\|}$, $\varepsilon > 0$ is arbitary value

$$\therefore \|V_m - V_n\| \le \gamma^n \|V_1 - V_0\| \frac{1}{1-\gamma} < \varepsilon$$

$\therefore$ the sequence V_i is Cauchy, and thus must have a fix point.

$$V^* = \lim_{n\to\infty} V_n = \lim_{n\to\infty} \beta V_{n-1} = \beta(\lim_{n\to\infty} V_{n-1}) = \beta V^*$$

$\therefore$ it conveges to a fix point.

unique: if it has two fix point. V_1, V_2 then.

$V_1 = \beta V_1$, $V_2 = \beta V_2$.

$\|\beta V_1 - \beta V_2\|_\infty = \|V_1 - V_2\|_\infty \leq \gamma \|V_1 - V_2\|_\infty$

$\because \gamma \in (0,1)$

$\therefore \|V_1 - V_2\| = 0$

$\therefore$ unique. fix point.

according to eq. 3.19 from the book,

$$\beta V_*(s) = \max_a E[R_{t+1} + \gamma V_*(S_{t+1}) | S_t = s, A_t = a]$$
$$= \max_{a \in A} q_{\pi_*}(s,a)$$
$$= V_*(s)$$

so this unique fixed point satisfied (or equivalent to) Bellman optimal equation.