ECEN 5018 Spring 2014 Midterm **Vince Coghlan**

1) One role for learning algorithms in distributed control is to guide the decisions of players in real time. However, in many settings players may not have the ability to select any particular action in their action set at any given time. For example consider the problem of multi-vehicle motion control where an agent's action set represents discrete spatial locations. Here, mobility limitations restrict the ability to traverse from one location to another in a given time period.

To formalize this notion of constrained action sets, consider the following process: Let $a(t-1)$ be the joint action at time $t-1$. With constrained action sets, the set of actions available to player i at time t is a function of his action at time $t-1$ and will be denoted as $C_i(a_i(t-1)) \subseteq \mathcal{A}_i$. For example, consider the following two player identical interest game with payoffs

	b_1	b_2	b_3
a_1	0,0	0,0	9,9
a_2	10,10	-10,-10	-10,-10

and constrained action sets that satisfy

$$C_1(a_1) = \{a_1, a_2\}, C_1(a_2) = \{a_1, a_2\}$$

$$C_2(b_1) = \{b_1, b_2\}, C_2(b_2) = \{b_1, b_2, b_3\}, C_2(b_3) = \{b_2, b_3\}$$

Standard log-linear learning assumes an agent can access any action at each iteration, i.e., $C_i(a_i) = \mathcal{A}_i$ for all actions $a_i \in \mathcal{A}_i$. The goal of this problem is to understand how mobility limitations impact the resulting behavior associated with log-linear learning. Here, we now consider two variations of log-linear learning which can accommodate constrained action sets.

- **Variation #1:** At each time t, the updating player i plays a stragey $p_i(t) \in \Delta(\mathcal{A}_i)$ where

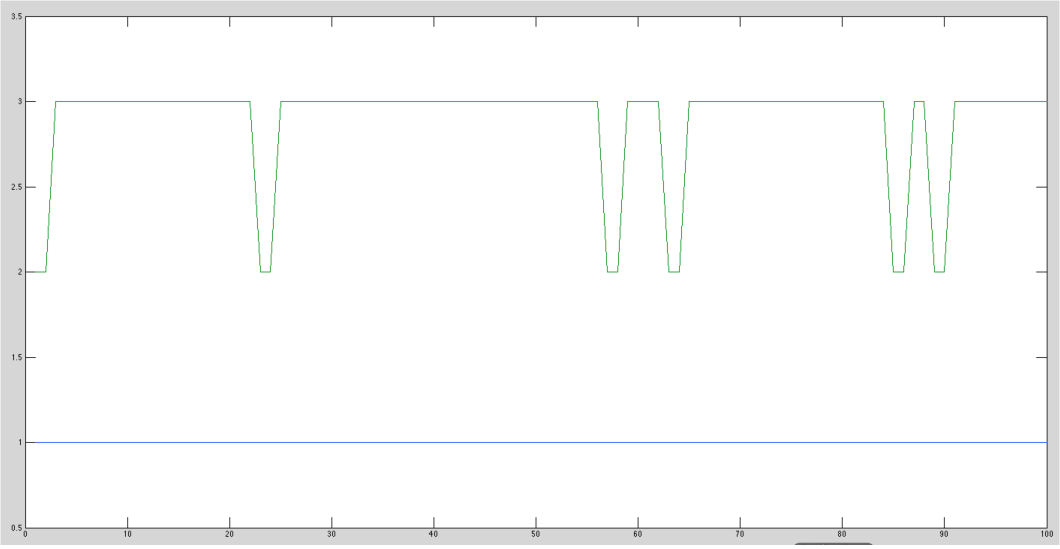
$$p_i^{a_i}(t) = \begin{cases} \frac{e^{\frac{1}{\tau}U_i(a_i, a_{-i}(t-1))}}{\sum_{\tilde{a}_i \in C_i(a_i)} e^{\frac{1}{\tau}U_i(\tilde{a}_i, a_{-i}(t-1))}} & \text{for any action } a_i \in C_i(a_i(t-1)) \\ 0 & \text{for any action } a_i \notin C_i(a_i(t-1)) \end{cases}$$

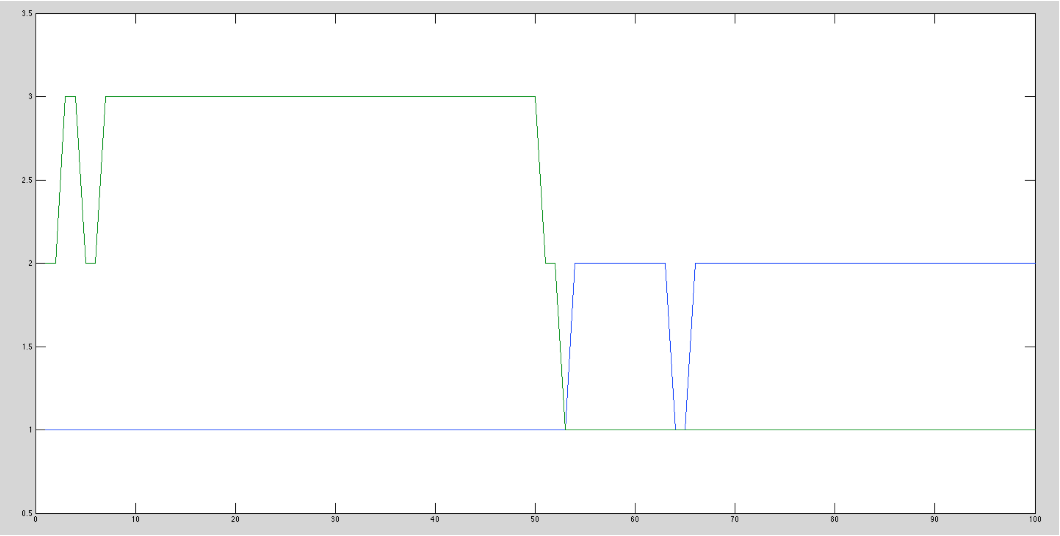
- **Variation #2:** At each time t, the updating player i selects one trial action $\tilde{a}_i$ (uniformly) from his constrained action set $C_i(a_i(t-1)) \subset \mathcal{A}_i$. Player i plays a strategy $p_i(t) \in \Delta(A_i)$ where

$$p_i^{a_i}(t) = \begin{cases} \frac{e^{\frac{1}{\tau}U_i(a_i, a_{-i}(t-1))}}{\sum_{\bar{a}_i \in \{a_i(t-1), \tilde{a}_i\}} e^{\frac{1}{\tau}U_i(\bar{a}_i, a_{-i}(t-1))}} & \text{for any action } a_i \in \{a_i(t-1), \tilde{a}_i\} \\ 0 & \text{for any action } a_i \notin \{a_i(t-1), \tilde{a}_i\} \end{cases}$$

1. Characterize the behavior of log-linear learning with Variation #1 as $\tau \to 0^+$ for the above example.

   In this strategy, each player picks based on each action that they can take at that time. It should be noted that for player 1, this will always act like the normal log-linear learning. For player 2, however, this can change. As $\tau \to 0$ the equilibria chosen depends on the initial position, since once you end up somewhere, you stay there. For example, these two plots were made when τ was relatively high ($\tau = 5$):

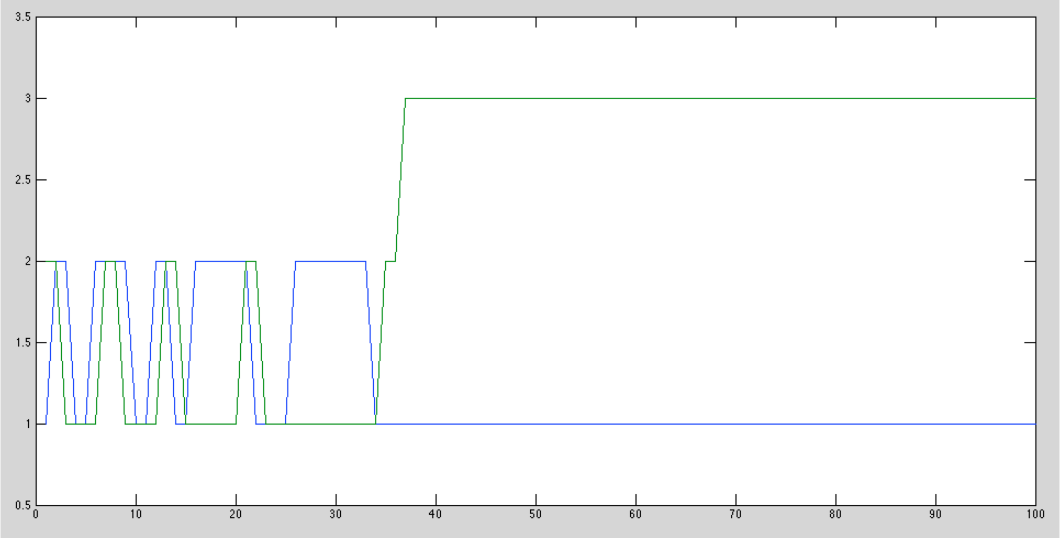




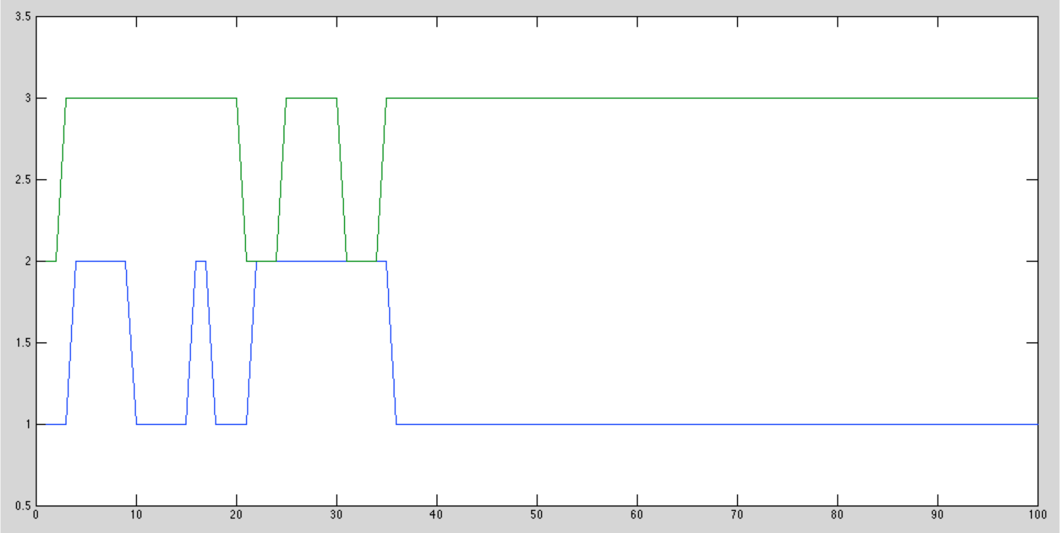
   These are the two equilibria of the system. Note that one can move from one to the other, but the probability of doing that is not just ϵ, but ϵ^2. As $\tau \to 0^+$ we can see the behavior approch one of these equilibrium and then stay there. Which equilibrium depends entirely on where it starts.

2. Characterize the behavior of log-linear learning with Variation #1 as $\tau \rightarrow 0^+$ for the above example.

   In this strategy, players pick uniformly over their distrobutions which path they want to analyze, then calculate the log-linear strategy of that action versus the action they took last turn. This strategy seems to play out exactly the same, quickly converging to one of the two NE. In fact, if I run each algorithm thousands of times, both of them converge to a_1b_3 about 70% of the time and a_2b_1 about 30% of the time, where both games are given completely random starting points. Additionally, almost every time variation #2 would take much less changes before it settled down. For instance, a typical run of variation #1 going from $\tau = 100$ to $\tau = 1$ would look like:



   Whereas a typical run for variation #2 would look like:



As you can see, there is a sort of inertia to variation # 2. This is because it takes into account the previous move. If you are anywhere on the board, you have a uniform probability (1/3 or 1/2) to stay there.

All of this code can be seen at the end of the test.

2) Consider the following class of resource allocation problems.

- A set of agents $N = \{1, 2\}$
- A finite set of resources $\mathcal{R} = \{r_1, r_2, r_3\}$
- An anonymous welfare function for each resource r of the form $W_r : \{0, 1, 2\} \rightarrow R$. The specific welfare functions are:

$$W_{r_1}(0) = 0,\ W_{r_1}(1) = v_1,\ W_{r_1}(2) = 1.5 \cdot v_1$$

$$W_{r_2}(0) = 0,\ W_{r_2}(1) = v_2,\ W_{r_2}(2) = 1.5 \cdot v_2$$

$$W_{r_3}(0) = 0,\ W_{r_3}(1) = v_3,\ W_{r_3}(2) = 1.5 \cdot v_3$$

  where $v_1, v_2, v_3 \geq 0$ but unknown.

- A finite action set of each player $\mathcal{A} \subseteq \mathcal{R}$. Note that an action $a_i \in \mathcal{A}_i$ corresponds to just a single resource $r \in R$. Let $\mathcal{A} := \mathcal{A}_1 \times \mathcal{A}_2$ represent the set of joint actions.
- A system-level welfare function of the form

$$W(a) = \sum_{r \in R} W_r(|a|_r)$$

  Where $|a|_r$ represents the number of players that chose resource r in the action profile a, i.e.,

$$|a|_r = |\{j \in N : r \in a_j\}|$$

The goal of this question is to understand how available utility design methodologies impact the efficiency of the resulting pure Nash equlibria.

(a) Consider the special case where $\mathcal{A}_1 = \mathcal{A}_2 = \{r_1, r_2\}$ (i.e., no resource r_3 for this specific problem); However, v_1 and v_2 are unknown. Suppose each agent is assigned marginal contribution utility. What is the price of anarchy and price of stability of the class of games induced by the set of feasable values $v_1, v_2 \geq 0$.

We can imagine a case where both players are on seperate routes. In order for player 1 to leave his resource, $v_1 < 0.5 \cdot v_2$. For player 2 to leave, $v_2 < 0.5 \cdot v_1$. Now assume that we are on the same route. We imagine a situation on the edge, where $v_1 = 0.5 \cdot v_2$. We would be at NE, but with a global welfare of:

$$W(a) = 1.5 \cdot v_2 = 1.5 \cdot 2 \cdot v_1 = 3 \cdot v_1$$

In this same game, if they are both on different routes, the global welfare would be:

$$W(a) = v_1 + v_2 = v_1 + 2 \cdot v_1 = 3 \cdot v_1$$

Meaning a PoA of:

$$\frac{3 \cdot v_1}{3 \cdot v_1} = 1$$

Well, the optimal path is the same as the NE, regardless of which of the two NE we are at in this game. We must now check on either side of thie inequality, first, if $v_1 < 0.5 \cdot v_2$, then both players will play v_2 with a welfare function of:

$$W(a) = 1.5 \cdot v_2$$

This is also the optimal, since the other payoff would be:

$$W(a) = v_1 + v_2 < v_2 + 0.5 \cdot v_2 \Rightarrow v_1 + v_2 < 1.5 \cdot v_2$$

This is also a PoA of 1, since the optimal is the NE. The only other case is when $v_1 > 0.5 \cdot v_2$ and $v_2 > 0.5 \cdot v_1$. The NE here is when the players are playing different resources, a welfare of:

$$W(a) = v_1 + v_2 > v_1 + \frac{1}{2} v_1 = 1.5 \cdot v_1$$

Again the NE is the optimal, a PoA of 1. We have thus found:

$$\text{PoA} = 1, \text{ PoS} = 1$$

(b) Consider the special case where $\mathcal{A}_1 = \{r_1, r_2\}$ and $\mathcal{A}_2 = \{r_2, r_3\}$; however, v_1, v_2, and v_3 are unknown. Suppose each agent is assigned the marginal contribution utility. What is the price of anarchy and price of stability of the class of games induced by the set of feasable values $v_1, v_2, v_3 \geq 0$

This is much like the last game, except that we can have very poor NE. For example, if $v_1 = \frac{1}{2} v_2$ and $v_3 = v_2$. In this case, player 1 could play r_1 and player 2 could play r_2. No player would move because his new location would simply be the same payoff as before. The optimal situation is if player 1 was on r_2 and player 2 was on r_3. That would provide a welfare of $v_2 + v_3 = 2 \cdot v_2$. The NE only provides a welfare of $v_1 + v_2 = 1.5 \cdot v_1$. This means:

$$\text{PoA} = \frac{4}{3}, \text{ PoS} = 1$$

PoS is 1 because in the last scenario, the optimum was another NE. There is no worse situation, this is as bad as it gets. This is because if v_1 was any higher, the PoA would still be positive, but it can only decrease. if it was any lower, player 1 would move to r_2. If v_2 was any higher, player 1 would move to it, if it was lower the PoA would drop. If v_3 were any higher, player 2 would move to r_3. If it were lower, PoA would drop. In all of these situations, we either move to a new NE that is optimum, or the PoA drops. Never do we increase PoA.

(c) Consider the special case where $\mathcal{A}_1 = \{r_1, r_2\}$ and $\mathcal{A}_2\{r_2, r_3\}$; however, v_1, v_2, and v_3 are unknown. Suppose each agent is assigned the Shapley value utility. What is the price of anarchy and price of stability of the class of games induced by the set of feasible values $v_1, v_2, v_3 \geq 0$.

We can imagine the same scenario as above. Now the center point is only worth $0.75 \cdot v_2$ for player 2, which means that he would rather play the full v_2 at r_3. This means that the PoS for this game is 1, because for at least one game, the optimal is played. Now imagine a different game, where $v_1 = v_3$ and $v_1 = \frac{4}{3}v_3$. This is the edge point of this game, since at this point each player sees the same payoff at either action. If they both play the same move then they will achieve a global welfare of $2v_3$. If the both play middle, the will get the same global welfare. If r_1 or r_3 was any larger, than that would be played by players 1 or 2 respectively. In this case the global welfare would increase by the increase in that route, but that would now be the optimal welfare, since both players playing r_2 would remain the same If r_1 or r_3 were to lower, than both players would play r_2 and would be achieving the optimal. If r_2 were to increase, both players would play r_2. If it were to decrease than both players would play their own respective chunks, which would be optimal.

In simple terms, the shapley value utility is based only on the amount of players and their action sets, not on what is currently being played. So there will always be only one equilibrium, and it will be optimal. If there are more than one equilibrium they will all have the same global welfare. We have:

$$\text{PoA} = 1, \text{ PoS} = 1$$

3) Two players are to chose between two movies. Player 1 (row) reads movie reads movie reviews and has a better idea which one is better. Player 2 (col) ignores movie reviews. The player preferences reflect a desire to see the better movie but also to go together. This scenario may be described as a Bayesian game as follows:

- States: ω_1 and ω_2
- State dependent payoff matricies:

$$\underset{\omega_1}{\begin{array}{c|c|c|} & A & B \\ \hline A & 3,3 & 2,0 \\ \hline B & 0,2 & 1,1 \\ \hline \end{array}} \qquad \underset{\omega_2}{\begin{array}{c|c|c|} & A & B \\ \hline A & 1,1 & 0,2 \\ \hline B & 2,0 & 4,4 \\ \hline \end{array}}$$

- Types & beliefs (i.e., $\Pr(\omega)$ given type) over the states $\{\omega_1, \omega_2\}$:
  - Player 1 (row) has 2 types, described by $t_1 = \tau_1(\omega)$ as follows:
    - $\tau_1(\omega_1) = \alpha$ with beliefs $\{2/3, 1/3\}$
    - $\tau_1(\omega_2) = \beta$ with beliefs $\{1/3, 2/3\}$
  - Player 2 (col) has only 1 type, described by $t_2 = \tau_2(\omega)$ as follows:
    - $\tau_2(\omega_1) = \tau_2(\omega_2) = (\alpha\beta)$ with beliefs $\{1/2, 1/2\}$

(a) Build the entire best response map for column player 2 (over pure strategies)
player 2 does not know that state, but believes that there is a 1/2 probability that we are in any of these states. We can find the payoffs for player 2 knowing this fact:

$$U_2(AA) = \frac{1}{2}3 + \frac{1}{2}1 = 2$$

$$U_2(AB) = \frac{1}{2}0 + \frac{1}{2}2 = 1$$

$$U_2(BA) = \frac{1}{2}2 + \frac{1}{2}0 = 1$$

$$U_2(BB) = \frac{1}{2}1 + \frac{1}{2}4 = 2\frac{1}{2}$$

From this we can see the best responses for player 2:

$$B_2(A) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \text{ and } B_2(B) = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Player 2 will always choose to see the same movie as player 1.

(b) Derive all Bayes-Nash equilibria (over pure strategies) Player 1 know some information about what state he is in, but he still does not have all the information. We can find his payoffs when given signal α:

$$U_{1\alpha}(AA) = \frac{2}{3}3 + \frac{1}{3}1 = 2\frac{1}{3}$$

$$U_{1\alpha}(AB) = \frac{2}{3}2 + \frac{1}{3}0 = 1\frac{1}{3}$$

$$U_{1\alpha}(BA) = \frac{2}{3}0 + \frac{1}{3}2 = \frac{2}{3}$$

$$U_{1\alpha}(BB) = \frac{2}{3}1 + \frac{1}{3}4 = 2$$

when given signal β:

$$U_{1\beta}(AA) = \frac{1}{3}3 + \frac{2}{3}1 = 1\frac{2}{3}$$

$$U_{1\beta}(AB) = \frac{1}{3}2 + \frac{2}{3}0 = \frac{2}{3}$$

$$U_{1\beta}(BA) = \frac{1}{3}0 + \frac{2}{3}2 = 1\frac{1}{3}$$

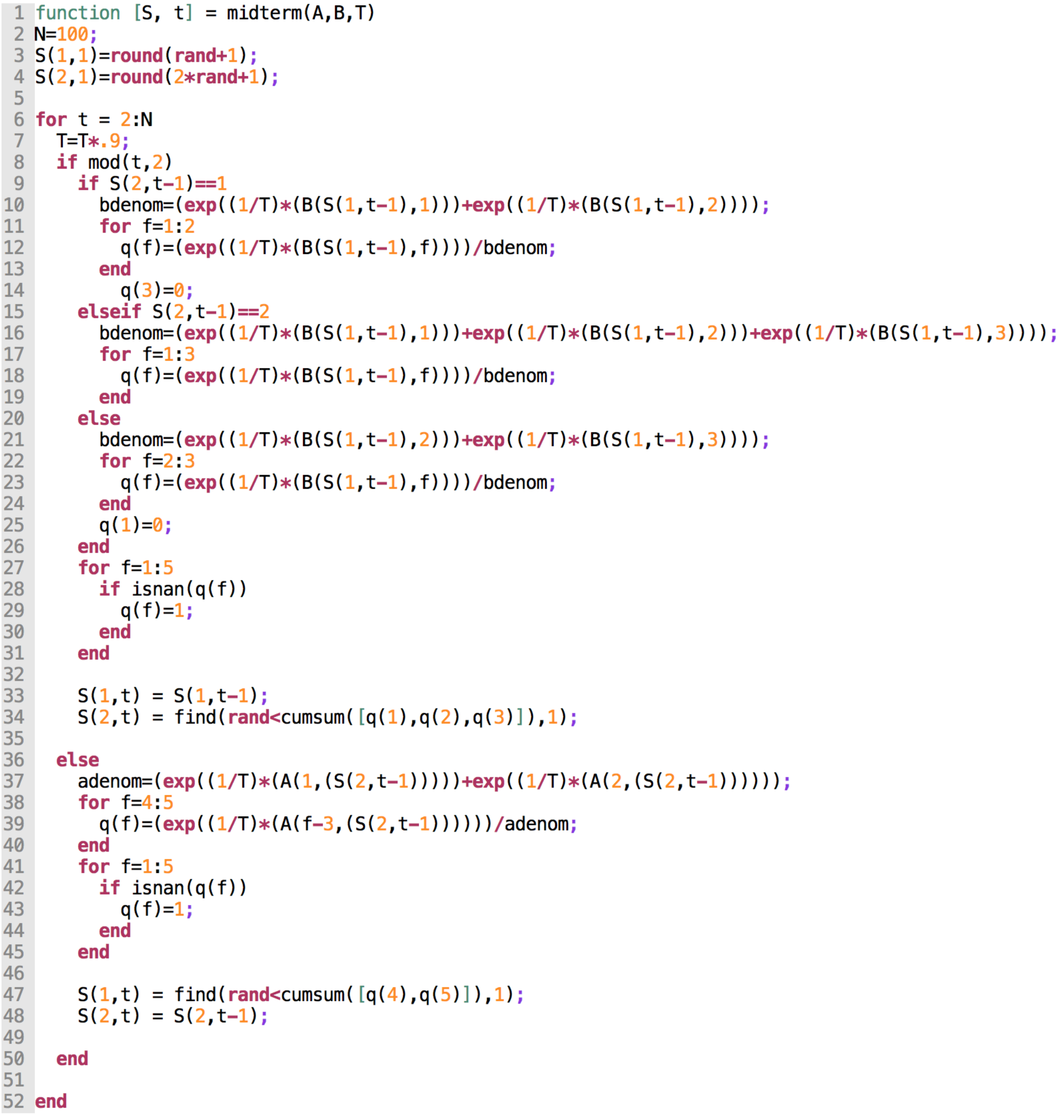
$$U_{1\beta}(BB) = \frac{1}{3}1 + \frac{2}{3}4 = 3$$

Alright, now we can move through action profiles to check if they are equilibria:

- {(A,A),A}: Player 2 is obviously happy, recieving a payoff of 2, player 1 also recieves a payoff of $(2\frac{1}{3}, 1\frac{2}{3})$ which is better than the alternatives of $\frac{2}{3}$ and $1\frac{1}{3}$ respectively.
- {(A,A),B}: Player 2 would rather play A.
- {(A,B),A}: Player 1 would rather play (A,A)
- {(A,B),B}: Player 1 would rather play (B,B)
- {(B,A),A}: Player 1 would rather play (A,A)
- {(B,A),B}: Player 1 would rather play (B,B)
- {(B,B),A}: Player 2 would rather play B.
- {(B,B),B}: Player 2 is happy, getting his highest possible payoff. Player 1 is getting $(2, 3)$, Which is better than $1\frac{1}{3}$ and $\frac{2}{3}$.

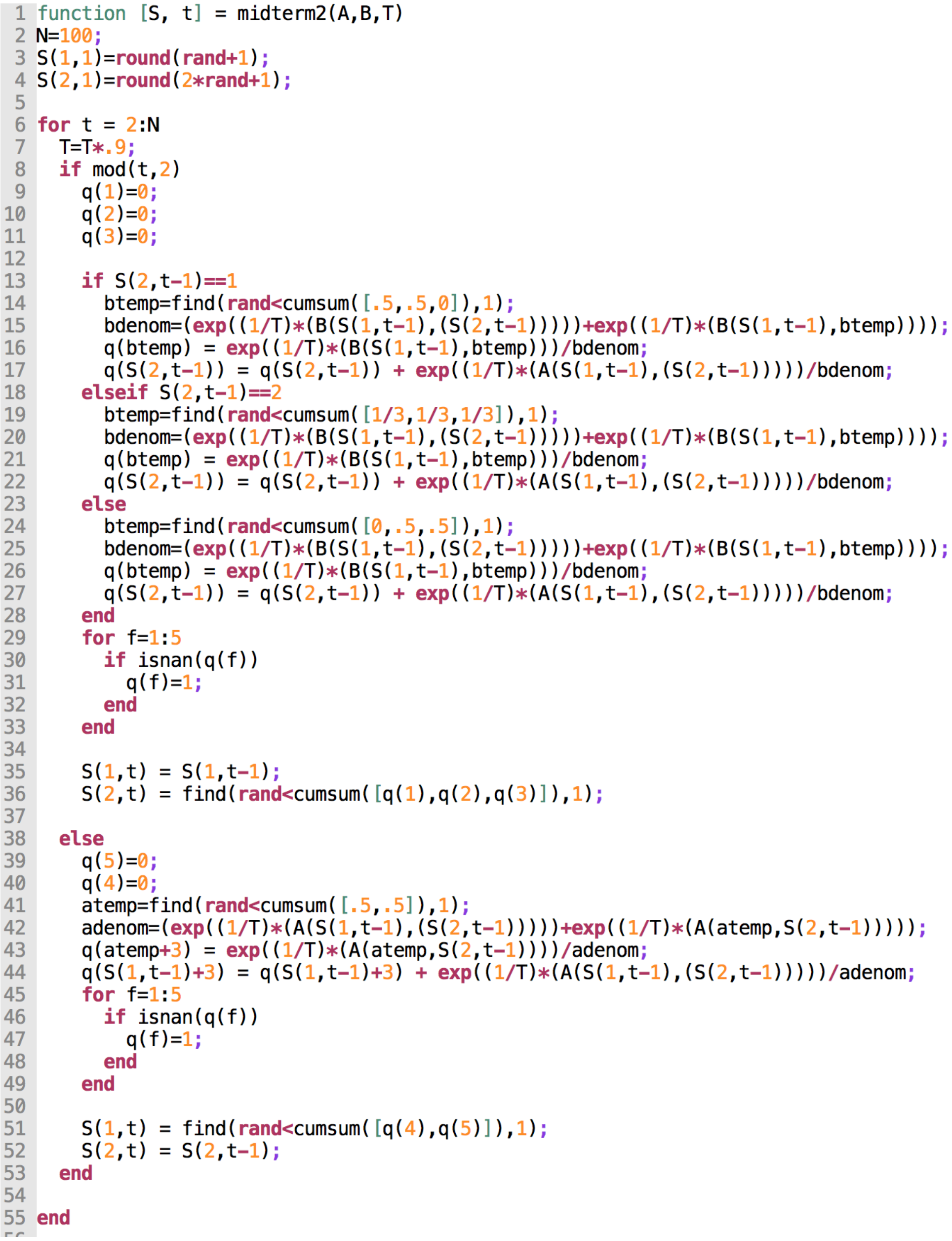
We have shown that the only two NE are {(A,A),A} and {(B,B),B}

Variation #1



```
function [S, t] = midterm(A,B,T)
N=100;
S(1,1)=round(rand+1);
S(2,1)=round(2*rand+1);

for t = 2:N
  T=T*.9;
  if mod(t,2)
    if S(2,t-1)==1
      bdenom=(exp((1/T)*(B(S(1,t-1),1)))+exp((1/T)*(B(S(1,t-1),2))));
      for f=1:2
        q(f)=(exp((1/T)*(B(S(1,t-1),f))))/bdenom;
      end
        q(3)=0;
    elseif S(2,t-1)==2
      bdenom=(exp((1/T)*(B(S(1,t-1),1)))+exp((1/T)*(B(S(1,t-1),2)))+exp((1/T)*(B(S(1,t-1),3))));
      for f=1:3
        q(f)=(exp((1/T)*(B(S(1,t-1),f))))/bdenom;
      end
    else
      bdenom=(exp((1/T)*(B(S(1,t-1),2)))+exp((1/T)*(B(S(1,t-1),3))));
      for f=2:3
        q(f)=(exp((1/T)*(B(S(1,t-1),f))))/bdenom;
      end
      q(1)=0;
    end
    for f=1:5
      if isnan(q(f))
        q(f)=1;
      end
    end

    S(1,t) = S(1,t-1);
    S(2,t) = find(rand<cumsum([q(1),q(2),q(3)]),1);

  else
    adenom=(exp((1/T)*(A(1,(S(2,t-1)))))+exp((1/T)*(A(2,(S(2,t-1))))));
    for f=4:5
      q(f)=(exp((1/T)*(A(f-3,(S(2,t-1))))))/adenom;
    end
    for f=1:5
      if isnan(q(f))
        q(f)=1;
      end
    end

    S(1,t) = find(rand<cumsum([q(4),q(5)]),1);
    S(2,t) = S(2,t-1);

  end

end
```

Variation #2



```
function [S, t] = midterm2(A,B,T)
N=100;
S(1,1)=round(rand+1);
S(2,1)=round(2*rand+1);

for t = 2:N
  T=T*.9;
  if mod(t,2)
    q(1)=0;
    q(2)=0;
    q(3)=0;

    if S(2,t-1)==1
      btemp=find(rand<cumsum([.5,.5,0]),1);
      bdenom=(exp((1/T)*(B(S(1,t-1),(S(2,t-1)))))+exp((1/T)*(B(S(1,t-1),btemp))));
      q(btemp) = exp((1/T)*(B(S(1,t-1),btemp)))/bdenom;
      q(S(2,t-1)) = q(S(2,t-1)) + exp((1/T)*(A(S(1,t-1),(S(2,t-1))))))/bdenom;
    elseif S(2,t-1)==2
      btemp=find(rand<cumsum([1/3,1/3,1/3]),1);
      bdenom=(exp((1/T)*(B(S(1,t-1),(S(2,t-1)))))+exp((1/T)*(B(S(1,t-1),btemp))));
      q(btemp) = exp((1/T)*(B(S(1,t-1),btemp)))/bdenom;
      q(S(2,t-1)) = q(S(2,t-1)) + exp((1/T)*(A(S(1,t-1),(S(2,t-1))))))/bdenom;
    else
      btemp=find(rand<cumsum([0,.5,.5]),1);
      bdenom=(exp((1/T)*(B(S(1,t-1),(S(2,t-1)))))+exp((1/T)*(B(S(1,t-1),btemp))));
      q(btemp) = exp((1/T)*(B(S(1,t-1),btemp)))/bdenom;
      q(S(2,t-1)) = q(S(2,t-1)) + exp((1/T)*(A(S(1,t-1),(S(2,t-1))))))/bdenom;
    end
    for f=1:5
      if isnan(q(f))
        q(f)=1;
      end
    end

    S(1,t) = S(1,t-1);
    S(2,t) = find(rand<cumsum([q(1),q(2),q(3)]),1);

  else
    q(5)=0;
    q(4)=0;
    atemp=find(rand<cumsum([.5,.5]),1);
    adenom=(exp((1/T)*(A(S(1,t-1),(S(2,t-1)))))+exp((1/T)*(A(atemp,S(2,t-1)))));
    q(atemp+3) = exp((1/T)*(A(atemp,S(2,t-1))))/adenom;
    q(S(1,t-1)+3) = q(S(1,t-1)+3) + exp((1/T)*(A(S(1,t-1),(S(2,t-1))))))/adenom;
    for f=1:5
      if isnan(q(f))
        q(f)=1;
      end
    end

    S(1,t) = find(rand<cumsum([q(4),q(5)]),1);
    S(2,t) = S(2,t-1);
  end

end
```