# Graph Network Analysis of Conway's Game of Life

Hakan Akgün

January 2024

## 0.1 Introduction

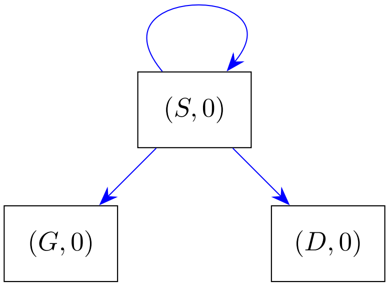
In this report, I'll explain how Conway's Game of Life (GoL) can be converted into a graph network and examined through graph theory tools by using a transition matrix. This formalism allows us to examine the statistical features of GoL from another perspective.

GoL is governed by the following rules:

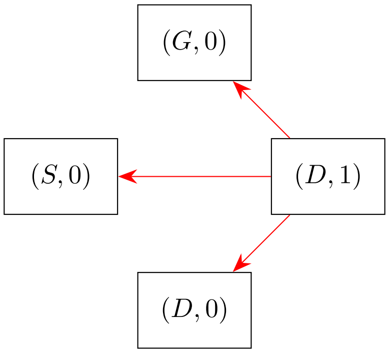
| | | |
|---|---|---|
| **Decay** | $m = 0, 1, 4, 5, 6, 7, 8$ | $-s$ |
| **Stability** | $m = 2$ | $0$ |
| **Growth** | $m = 3$ | $1 - s$ |

Table 1: Analysis of Conway's Game of Life States

These rules can be represented as a network. For example, a state 0 with 2 neighbors (m=2), i.e., stable, can be represented by the following connections:



Or a state 1 with 4 neighbors (m=3) can be represented by the following connections:



Via representing decay edges red, growth edges green, stability edges blue we acquire the following graph:

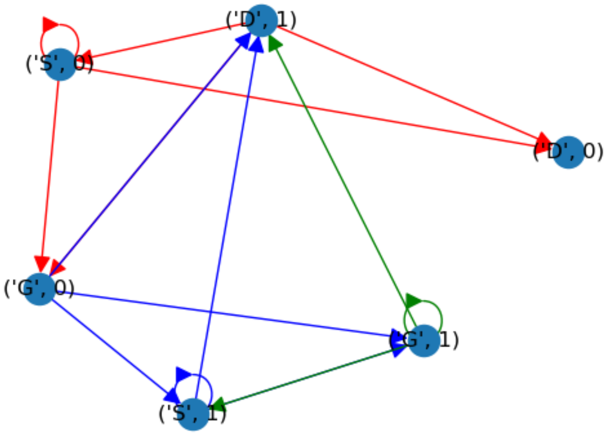
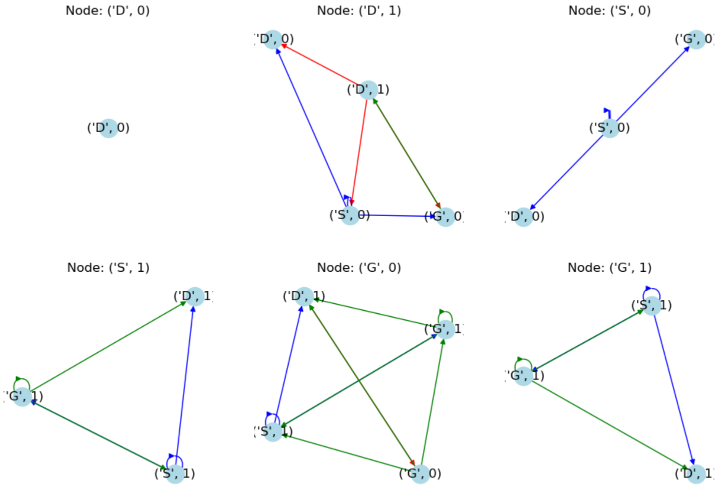


Figure 1: The first index represents the change that it will undergo in the next state (D: Decay, G: Growth, S: Stability), and the second index represents whether it's in state 0 or 1.

If we want to examine connections as (change,state) pairs, we acquire the following graphs:

This provides us with the representations that a state can undergo. However to make quantitaive analysis on these networks we need occupation probabilities. Here mean field approximations can help us:



## 0.2 Mean Field analysis

In our analysis, we approximate the network states probabilistically. Given that we have $n$ possible neighbors (where $n = 8$ in our case), the probability of a node being occupied by $k$ states is calculated using the binomial probability formula:

$$P(m = k) = \binom{n}{k} \cdot p^k \cdot (1-p)^{n-k}$$

Where:

- $P(m = k)$ is the probability of a node being occupied by $k$ neighbors.
- $\binom{n}{k}$ represents the binomial coefficient, which calculates the number of ways to choose $k$ neighbors out of $n$ possible neighbors.
- $p$ is the probability of being in state 1 (which is determined by the initial value of the uniform distribution since we start our grid with this uniform distribution).
- $1 - p$ is the probability of being in state 0.

This directly calculates probabilisticly being in states (D,G,s). By multiplying again with probability of having state 0 and one we get:

$$P(m = k, S = 1) = p \cdot \binom{n}{k} \cdot p^k \cdot (1-p)^{n-k}$$

&

$$P(m = k, S = 0) = (1 - p) \cdot \binom{n}{k} \cdot p^k \cdot (1 - p)^{n-k}$$

By using this approximation all state probabilities are calculated via following code block:

Listing 1: Calculation of Node Probability

```
if cond == 'G':
    % Growth condition
    binom_prob = binomial_p(8, 3, p)
elif cond == 'S':
    % Stability condition
    binom_prob = binomial_p(8, 2, p)
elif cond == 'D':
    % Decay condition
    binom_prob = 1 - (binomial_p(8, 3, p) + binomial_p(8, 2, p))

if state == 1:
    return binom_prob * p
elif state == 0:
    return binom_prob * (1 - p)
```

For each initial probability value(density in initial uniform distribution), we acquire set of values for each state. For example when we set p=0.6, we acquire the following state values:

| Condition | State 0 | State 1 |
|---|---|---|
| S | 0.017 | 0.025 |
| G | 0.050 | 0.074 |
| D | 0.334 | 0.501 |

Table 2: Node Probabilities for p=0.6

This means probability that being in a node can be approximated via explained mean field approximation.

As a simplified model for examining the system dynamics, we can represent the probability of transitioning between states as the probability of reaching the target node. For example, the transition probabilities from state $S_0$ to states $S_0$, $G_0$, and $D_0$ (i.e., all possible transitions from $S_0$) are respectively $\frac{0.017}{N}$, $\frac{0.050}{N}$, and $\frac{0.334}{N}$, where $N$ is the normalization factor given by $N = 0.017 + 0.050 + 0.334$ in this case.

By making this assumption, we can construct a directed graph in which transition probabilities are represented on the edges. This allows us to visualize and analyze the potential transitions between states within the system.

Later on we will use these graphs to find different features but let's first examine the corresponding matrix formulation for these graphs.

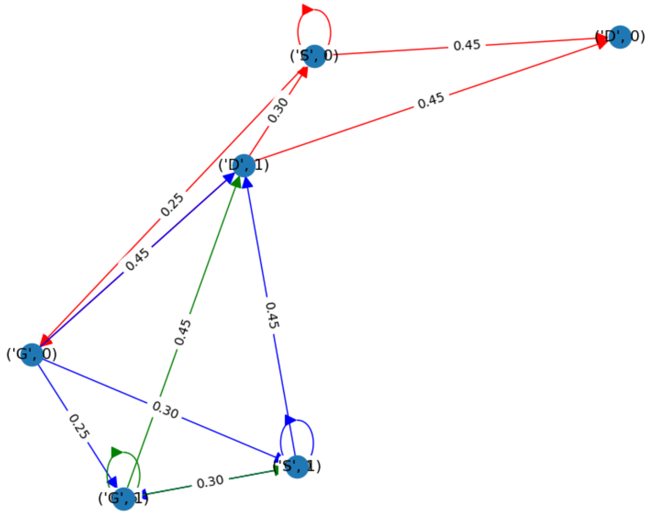


Figure 2: Each edge represents transition probabilities from given node.

## 0.3 Matrix formulation:

We can model transition probabilities from nodes with a transition matrix. In this context, the transition matrix corresponds to the probability of transitioning from one state to another. The vector that we multiply with the transition matrix represents the initial probabilities of being in those states. For example, updating the initial state once corresponds to the following relation

$$\begin{array}{c|cccccc} & \text{D0} & \text{D1} & \text{S0} & \text{S1} & \text{G0} & \text{G1} \\ \hline \text{D0} & T_{1\_D} & 0 & T_{1\_S} & 0 & T_{1\_G} & 0 \\ \text{D1} & 0 & T_{2\_D} & 0 & T_{2\_S} & 0 & T_{2\_G} \\ \text{S0} & T_{3\_D} & 0 & T_{3\_S} & 0 & T_{3\_G} & 0 \\ \text{S1} & 0 & T_{4\_D} & 0 & T_{4\_S} & 0 & T_{4\_G} \\ \text{G0} & 0 & T_{5\_D} & 0 & T_{5\_S} & 0 & T_{5\_G} \\ \text{G1} & 0 & T_{6\_D} & 0 & T_{6\_S} & 0 & T_{6\_G} \end{array} \begin{bmatrix} I_{D0} \\ I_{D1} \\ I_{S0} \\ I_{S1} \\ I_{G0} \\ I_{G1} \end{bmatrix} = \begin{bmatrix} T_{1\_D} I_{D0} + T_{1\_S} I_{S0} + T_{1\_G} I_{G0} \\ T_{2\_D} I_{D1} + T_{2\_S} I_{S1} + T_{2\_G} I_{G1} \\ T_{3\_D} I_{D0} + T_{3\_S} I_{S0} + T_{3\_G} I_{G0} \\ T_{4\_D} I_{D1} + T_{4\_S} I_{S1} + T_{4\_G} I_{G1} \\ T_{5\_D} I_{D1} + T_{5\_S} I_{S1} + T_{5\_G} I_{G1} \\ T_{6\_D} I_{D1} + T_{6\_S} I_{S1} + T_{6\_G} I_{G1} \end{bmatrix}$$

Values in **T** can be acquired via our aforementioned toy model and nth state of this representation corresponds to $\mathbf{T^n}\vec{I} = \vec{I}_n$. Since $T_{ij} \neq T_{ji}$, the matrix **T** is non-Hermitian and asymmetric.

### 0.4 Network analysis of graph formalism

The good part of network representations is that they equip us with an unseen analysis through mean field approximation. One of the techniques used in graph theory is centrality measurement. Centrality measurements tell us about the importance of nodes in different aspects; by understanding these aspects, we can gain a deeper understanding of GOL states. Here, we'll initially examine two centrality measurements (centrality measurements rarely use edge weights, so other centrality measurements are not as useful in our context):

- Betweenness Centrality

  In graph theory, betweenness centrality is a measure of centrality based on shortest paths. For every pair of vertices in a connected graph, there exists at least one shortest path between the vertices such that the sum of the weights of the edges (for weighted graphs) is minimized. The betweenness centrality for each vertex is the number of these shortest paths that pass through the vertex. However, since it looks for minimum paths, we will adjust our weights in betweenness measurement as weight $= \frac{1}{transitionprobability}$ , since we want to maximize probabilities while minimizing distances of shortest paths. High betweenness centrality, in our case, will correspond to which states most probably will be visited to go to the next states. This also corresponds to the notion of "variety" for state space. I think that states with high betweenness scores will be responsible for the "variety" of the state space (since they provide the most transitions). Therefore, by using this score, we can quantify the variety of the state space.

- PageRank Centrality

  PageRank centrality is a measure used in network analysis that ranks the importance of nodes within a graph. It is based on the idea that connections to high-importance nodes contribute more to the importance of a node than equal connections to low-importance nodes (by using in-degree centrality, which is more important for our assumptions since it's about the probability of ending up in a state). It's a variant of eigenvector centrality, but eigenvector centrality works better for undirected graphs, while PageRank works better for directed ones. The PageRank score can be thought of as the probability that we come across a specific website when we are randomly clicking on webpage links on the internet. Similarly, in our case, given that the mean field approximation describes the probabilities accurately, the PageRank score describes the probability of seeing a certain state when we look at a randomly selected point, blindfolded in the long run! (The 'in the long run' part is important because this implies that PageRank scores are not simple initial probability scores but they are long-run results acquired by calculations of the stable eigenvectors of the network - Remember, it's a variant of eigenvector score; we haven't used eigenvector score directly because it's for undirected graphs, unlike our directed graphs).

Using these centrality scores for probability of being in state 1=0.6 in our mean field approximation, we acquire following scores:

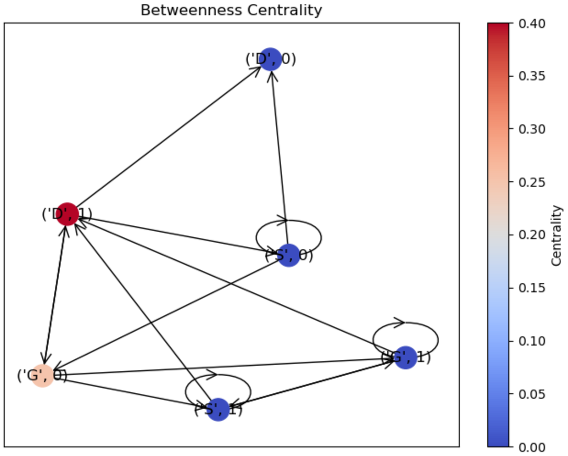


Figure 3: For example, by looking at this graph, we can claim that most of the states come from the $(D, 1)$ state before the last update. And in the next state, the number of different states coming from the $(D, 1)$ state will be the most. Similarly, the second node providing this variety will be the $(G, 0)$ state.

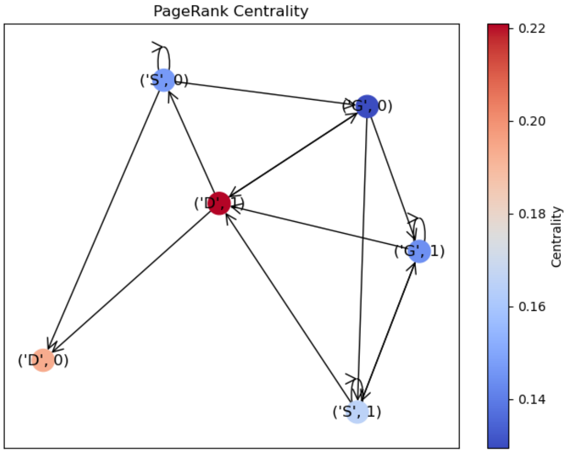


Figure 4: For example, by looking at this graph, we can claim that when selecting a random state in the whole state space, we will be seeing $(D, 1)$ with the highest probability. Then the second most probable state for this statement is $(D, 0)$.

These graphs are for $p = 0.6$ probability. Now, we should look for the whole $p \in [0, 1]$ space to make further quantitative and qualitative analyses. When we acquire and plot centrality measures for individual nodes over this range, we obtain the following graphs:

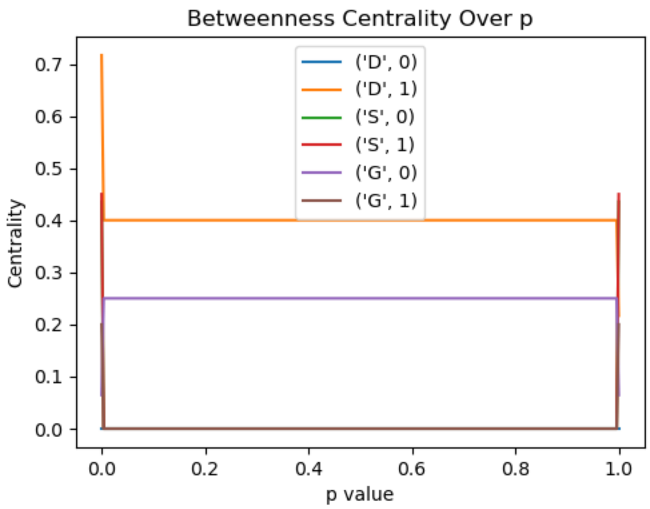
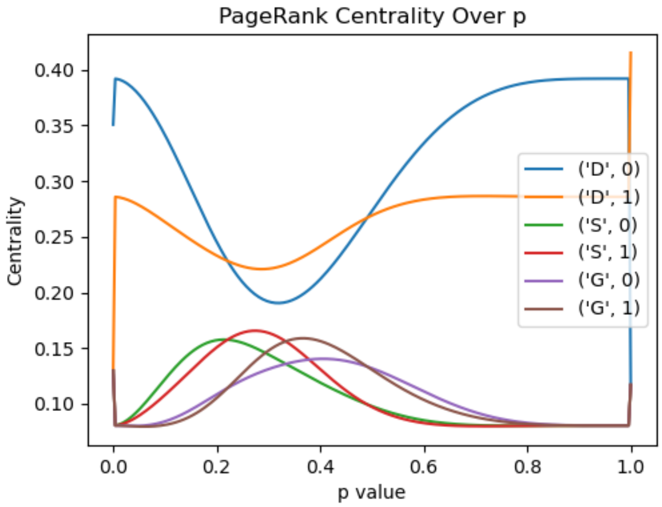


Figure 5: As can be seen, regardless of the initial probability value, the states providing variety are always the first $(D, 1)$ and the second $(G, 0)$.



Note:Maybe this notion can be further examined in the context of ”halting problem”

Figure 6: As can be seen, the $p$ value changes the PageRank score for all states, and after $p = 0.8$, all states become decaying states. Therefore, it can possibly be claimed that after $p = 0.8$ the state space is ”inactive” (either stable periodic structures or 0 states). On the other hand, it’s interesting to see that the minimums of D states and the maximums of S and G states are very close. This point, I think, indicates the point of ”highest variety,” and the probability of transitioning to the ”inactive phase” is very low.

Note:Maybe this notion can be further examined in the context of ”halting problem”

Comparing our centrality predictions with actual evolutions of the states over probabilities and times, we acquire the following graphs:

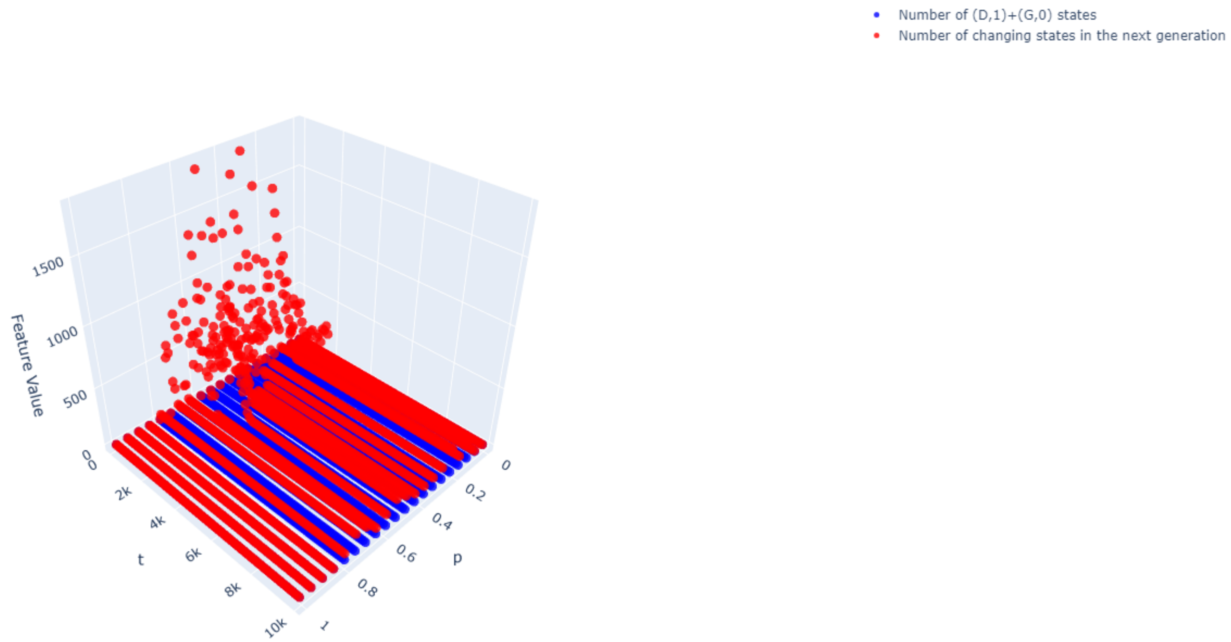


Figure 7: In this plot, we have plotted the number of $D_1 + G_1$ states and the number of changes of states, where a change, as an example, is taken as $D_1$ to $D_0$, $S_0$ to $S_0$ - not only $0 \rightarrow 1$ transitions, but all 6 state transitions are taken as a "change." As can be seen, change happens only when $D_1 + G_1$ is greater than 0 for a given point in time, which aligns with the prediction of betweenness centrality.

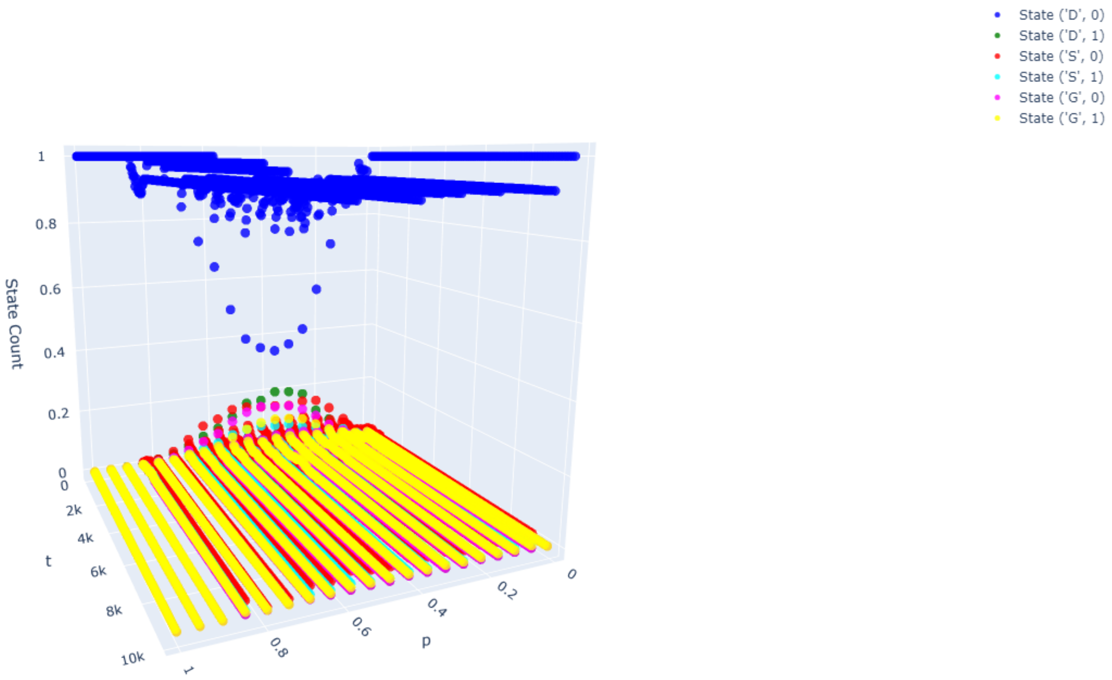


Figure 8: In this plot, we have plotted the number of different state times over time. This doesn't align with the prediction of the PageRank algorithm. Assuming my interpretation is correct for the algorithm, I believe that the mean field approximation can be improved to acquire better (closer to true transition values) edge probabilities and long-run behavior for states.

# 1 Logistic Game of Life Extansion

In logistic GOL states are no longer discrete but still we can model our transition matrix like the following:

$\mathbf{p} = \begin{bmatrix} p \\ 1-p \end{bmatrix}$ (Probability Vector where upper row is for state 1 & and bottom row is for state 0 weight)

$$D = \begin{bmatrix} 1-\lambda & 0 \\ \lambda & 1 \end{bmatrix} \qquad \text{(Decay Matrix)}$$

$$S = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad \text{(Stability Matrix)}$$

$$G = \begin{bmatrix} 1 & \lambda \\ 0 & 1-\lambda \end{bmatrix} \qquad \text{(Growth Matrix)}$$

Which implies,

$$\to D\mathbf{p} = \begin{bmatrix} p-\lambda p \\ 1-p+\lambda p \end{bmatrix}$$

$$\to S\mathbf{p} = \begin{bmatrix} p \\ 1-p \end{bmatrix}$$

$$\to G\mathbf{p} = \begin{bmatrix} p+\lambda-\lambda p \\ (1-\lambda)(1-p) \end{bmatrix}$$

Now one possible way to get vector describing weight and probability of state space D0,D1,S0,S1,G0,G1 is combining $D\mathbf{p}$,$S\mathbf{p}$,$G\mathbf{p}$ vectors with probabilities of being in states D,G,S i.e :

$$\mathbf{p}' = \begin{bmatrix} p_{D,0} \\ p_{D,1} \\ p_{S,0} \\ p_{S,1} \\ p_{G,0} \\ p_{G,1} \end{bmatrix} = \begin{bmatrix} (p-\lambda p)(8-(t_3-t_1)) \\ (1-p+\lambda p)(8-(t_3-t_1)) \\ p(t_2-t_1) \\ (1-p)(t_2-t_1) \\ (p+\lambda-\lambda p)(t_3-t_2) \\ (1-\lambda)(1-p)(t_3-t_2) \end{bmatrix}$$

A good question at this point is that how we can acquire our network transition matrix from this probability vector $\mathbf{p}'$. I believe it's plausible to use tensor product at this point. Since probability of transitioning from D0 to G1 for example can be described as probability of being in state D0 initially* probability of being in G1 in the next state. This exactly corresponds to following tensor product:

$$\mathbf{p}'\mathbf{p}'^{\top} = \begin{bmatrix} p_{D,0} \\ p_{D,1} \\ p_{S,0} \\ p_{S,1} \\ p_{G,0} \\ p_{G,1} \end{bmatrix} \begin{bmatrix} p_{D,0} & p_{D,1} & p_{S,0} & p_{S,1} & p_{G,0} & p_{G,1} \end{bmatrix}$$

$$
= \begin{array}{c|cccccc}
 & \text{D0} & \text{D1} & \text{S0} & \text{S1} & \text{G0} & \text{G1} \\
\hline
\text{D0} & p_{D,0}^2 & p_{D,0}p_{D,1} & p_{D,0}p_{S,0} & p_{D,0}p_{S,1} & p_{D,0}p_{G,0} & p_{D,0}p_{G,1} \\
\text{D1} & p_{D,1}p_{D,0} & p_{D,1}^2 & p_{D,1}p_{S,0} & p_{D,1}p_{S,1} & p_{D,1}p_{G,0} & p_{D,1}p_{G,1} \\
\text{S0} & p_{S,0}p_{D,0} & p_{S,0}p_{D,1} & p_{S,0}^2 & p_{S,0}p_{S,1} & p_{S,0}p_{G,0} & p_{S,0}p_{G,1} \\
\text{S1} & p_{S,1}p_{D,0} & p_{S,1}p_{D,1} & p_{S,1}p_{S,0} & p_{S,1}^2 & p_{S,1}p_{G,0} & p_{S,1}p_{G,1} \\
\text{G0} & p_{G,0}p_{D,0} & p_{G,0}p_{D,1} & p_{G,0}p_{S,0} & p_{G,0}p_{S,1} & p_{G,0}^2 & p_{G,0}p_{G,1} \\
\text{G1} & p_{G,1}p_{D,0} & p_{G,1}p_{D,1} & p_{G,1}p_{S,0} & p_{G,1}p_{S,1} & p_{G,1}p_{G,0} & p_{G,1}^2
\end{array}
$$

This can be interpreted as our transition matrix, and a corresponding network can be created. It should be noted that in this formalism, due to the continuity of the states, a mean field approximation of states necessarily returns a Hermitian matrix. This is because given $p_{i,j} * p_{j,i} = p_{j,i} * p_{i,j}$ meaning $\mathbf{p'_{i,j}} = \mathbf{p'_{j,i}}$. Since our transition matrix is Hermitian for this formalism, the directedness nature of the network disappears, and we get equipped with further centrality measures. (Undirected graphs have more centrality measures, including Pagerank and Betweennes; we can investigate these further later if desired. However, remember that the PageRank algorithm works better for directed graphs as a variation of eigenvector centrality. Therefore, we'll be using eigenvector centrality in logistic GOL, since our network is undirected.)

Now, by using these connections, we can weight our edges in our network and observe how centrality measures change with the $\lambda$ parameter:

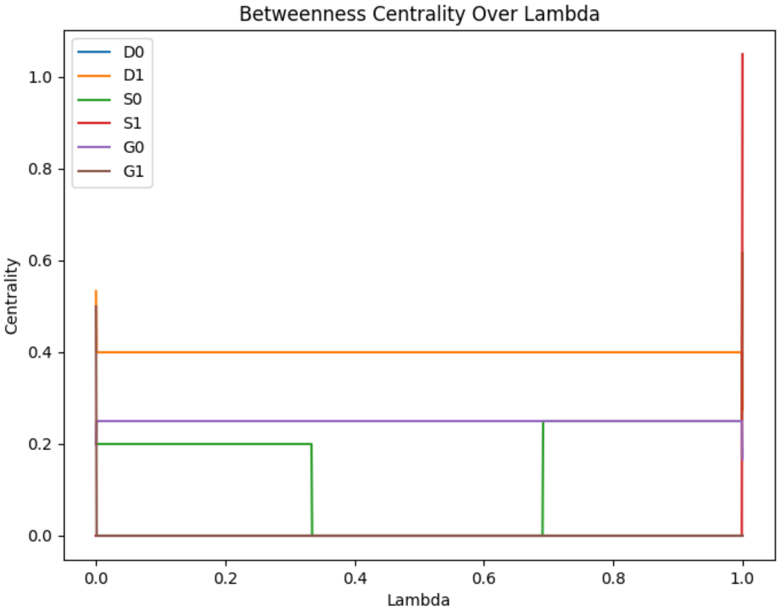


Figure 9: These results can be interpreted similarly to the previous ones. We have the $D1$ and $G0$ nodes as our main providers of variety, but this time the $S0$ state is also distinct. It's interesting to note the step-like behavior of the $S0$ state in this context.

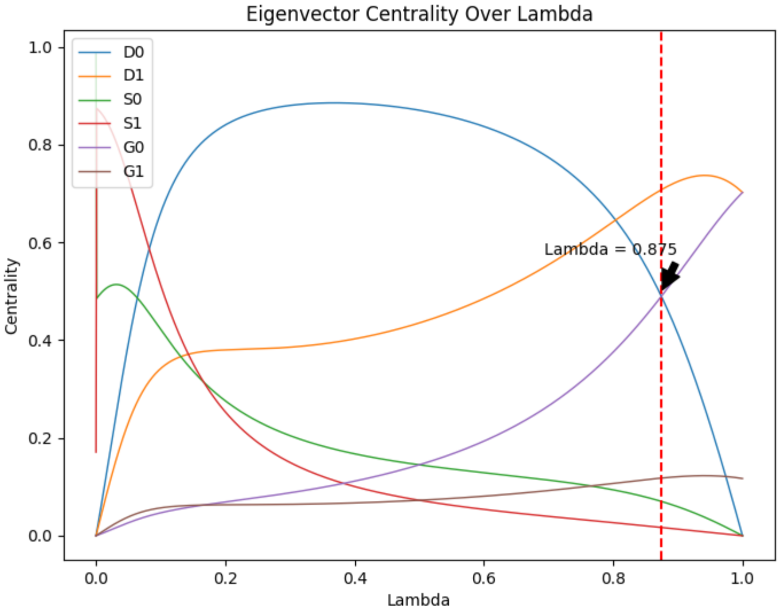


Figure 10: Interestingly, the transition point that Prof. Seymur described in his paper, can be find with this methodology and observed in the (G,0) and (D,0) plots. After this point, $G0$ and $D1$ become the only dominant states. This can be interpreted as the emergence of periodic and still-life structures, since these structures correspond to periodic occurrences of $G0$ and $D1$ states.

These analyses are performed for lambda parameters, but they can be easily generalized for also interval parameters t1,t2, and t3.

## 2 Further questions about matrix formalism:

Q1: Can we apply quantum mechanics formalism to find stable states? By our transition matrices we acquire time evolution operators of the matrices, can we find a connection between stable configurations and transition operator for any given initial vector. This is analogous to finding stable energy(minimum energy states) configurations of the system by knowing the time evolution operator in QM.

Q2: If this is possible, can we connect minimum energy states-stable states to halting problem? Given that low energy states will remain and these stable states will correspond to minimum activity states.